SHARP®

COMPUTER TASCABILE

MODELLO

PC-1403

MANUALE DI ISTRUZIONI

CONTENTO

Page

NOTE INTRODUTTIVE 1

CAPITOLO 1. USO DI QUESTO MANUALE 2

CAPITOLO 2. INTRODUZIONE 4
- Descrizione del sistema 4
- Funzionamento dei tasti e dell'interruttore 5
- Modi 6
- Tasto ALL RESET 9
- Controll del contrasto 10
- Sostituzione delle Batterie 10

CAPITOLO 3. USO COME CALCOLATRICE 13
- Accensione 13
- Spegnimento 13
- Auto off (Spegnimento automatico) 13
- Tasti usati per i calcoli nel modo CAL 14
- Come si legge il visore 18
- Calcoli 20
- Calcoli scientifici nel modo CAL 23
- Uso delle parentesi 27
- Posizione decimali 27
- Livelli di priorità 28
- Calcolo statistico 34
  1. Calcolo statistico con una variabile 35
  2. Statistica a due variabili e regressione linerare 37
- Gamma di calcolo 41
- Funzione di calcolo matriciale 44
- Calcolo manuale in BASIC 56
- Come eseguire manualmente i calcoli 56
- Richiamo dei dati immessi 57
- Errori 61
- Calcoli in serie 62
- Numeri negativi 63
- Calcoli composti e parentesi 64
- L'uso delle variabili nei calcoli 65
- Calcoli in catena 66
- Messaggio di errore 67
- Notazione scientifica 67

Limiti 68
Funzione ultimo risultato 69
Lunghezza delle formule 70
Calcoli scientifici nel modo BASIC 70
Funzione calcolo diretto 76
Ordine di priorità nei calcoli manuali 77

CAPITOLO 4. CONCETTI E TERMINI DEL BASIC 78
Le stringhe costanti 78
Numeri esadecimali 78
Le variabili 79
Variabili fisse 80
Variabili semplici 81
Matrici di variabili 82
Variabili in forma di A( ) 85
Espressioni 87
Operatori numerici 87
Espressioni stringa 88
Operatori logici/espressioni di relazione 88
Espressioni logiche 89
Parentesi e precedenze 91
Modo RUN 92
Funzioni 92

CAPITOLO 5. PROGRAMMAZIONE 93
Programmi 93
Dichiarazioni (statement) in BASIC 93
Numero di riga 93
I verbi in BASIC 94
I comandi in BASIC 94
Modi di funzionamento 94
Cominciamo a programmare il computer 95
Esempio N.1 – Inseriamo ed elaboriamo un programma 95
Esempio N.2 – Revisione del programma 96
Esempio N.3 – L'uso delle variabili nel programma 98
Esempio N.4 – Programmi più complessi 101
Permanenza dei programmi nella memoria del computer 102

CAPITOLO 6. SCORICIATOIE 103
Il tasto DEF ed i programmi etichettati 103
Mascherina 104

CAPITOLO 7. USO DELL'INTERFACCIA STAMPANTE/REGISTRATORE A CASSETTE CE-126P 105
Uso della stampante 105

Uso dell'interfaccia registratore a cassette 107
Registratore a cassette 107
Uso dell'interfaccia e del registratore 109
Note sul nastro 112

CAPITOLO 8. IL BASIC 113
Comandi 117
Verbi 134
Le funzioni 181
Le pseudovariabili 181
Funzioni numeriche 183
Funzioni stringa 189

CAPITOLO 9. RICERCA DEI GUASTI 191
Problemi con la macchina 191
Controllo dei programmi 192

CAPITOLO 10. MANUTENZIONE 194

APPENDICI
APPENDICE A Messaggi di errpre 195
APPENDICE B Tavola dei codici ASCII 198
APPENDICE C I formati di visualizzazopne 200
APPENDICE D Valutazione delle espressiono e priorità degli operatori 204
APPENDICE E Funzioni dei tasti nel BASIC 206
APPENDICE F Catattrristiche 212
APPENDICE G Uso di ptogrammi scritti con altri modelli PC 214

ESEMPI DI PROGRAMMAZIONE 220

INDICE 242

NOTE INTRODUTTIVE

Benevenuti nel mondo dei possessori **SHARP!**

Poche industrie al mondo sono in grado di tenere il passo con la rapida crescita e gli avanzamenti tecnologici realizzati nel campo dei personal computer. Gli elaboratori che fino a poco tempo fa avrebbero riempito una stanza di notevoli dimensioni, richiesto un esperto per la programmazione e che sarebbero costati migliaia di dollari, stanno ora nel palmo della mano, vengono facilmente programmati, e costano talmente poco da poter essere alla portata praticamente di chiunque.

Il vostro nuovo **computer** è stato realizzato secondo gli ultimissimi ritrovati di questa rivoluzione tecnologica ed incorpora molte funzioni avanzate:

* CALCOLATRICE SCIENTIFICA — Prima del **computer**, era normale usare due strumenti per due compiti diversi, il calcolo scientifico (inclusa la statistica) e l'elaborazione. Ora però é sufficiente un solo strumento. Il **computer** funziona sia da calcolatrice scientifica che da elaboratore tascabile in quanto incorpora molte funzioni scientifiche preprogrammate qiù i tasti di comando Basic per una semplice programmazione.
* PROTEZIONE DELLA MEMORIA — il **computer** ricorda i programmi e le variabili memorizzate anche quando viene spento.
* L'alimentazione a batteria consente una effettiva portatilità.
* La funzione AUTO POWER OFF (spegnimento automatico) preserva le batterie spegnendo la macchina dopo un certo tempo di inattività.
* Una versione ampliata del Basic consente la visualizzazione e la stampa secondo formati da voi predeterminati, matrici bidimensionali, stringhe di lunghezza variabili e molte altre funzioni avanzate.
* Viene fornita un interfaccia per la stampante e il registratore a cassette CE-126.

Congratulazioni per essere entrati in un nuovo mondo eccitante e piacevole. Siamo certi che considererete questo acquisto uno dei più saggi che abbiate mai fatto. Il **computer SHARP** è uno strumento potente, progettato per soddisfare le vostre esigenze specifiche in qualsiasi settore: matematico, scientifico, ingegneristico, commerciale e personale. Con il **computer SHARP** potete cominciare ORA ad ottenere le soluzioni che vi serviranno domani!

CAPITOLO 1
USO DI QUESTO MANUALE

Questo manuale è stato studiato per presentarvi le capacità e le funzioni del vostro **computer** e per servire da valido strumento di consultazione. Sia che siate un "principiante" che un "veterano" con i calcolatori, dovete imparare a conoscere il **computer** leggendo e lavorando sui capitoli 2–6.

* Il capitolo 2 descrive le caratteristiche fisiche del **computer**.
* Il capitolo 3 dimostra l'uso del **computer** come calcolatrice scientifica.
* Il capitolo 4 definisce alcuni termini e concetti essenziali per la programmazione in BASIC e dà alcune considerazioni speciali su questi concetti spplicati al **computer**.
* Il capitolo 5 vi introduce alla programmazione in Basic sul **computer**, illustrandovi come inserire, correggere ed eseguire i programmi.
* Il capitolo 6 tratta alcuni accorgimenti che rendono più facile e più gradevole l'uso del vostro nuovo computer.

Il capitolo 8 tratta tutti i verbi, i comandi e le funzioni del Basic in ordine di raggruppamento alfabetico, per una più facile consultazione.

I programmatori esperti in Basic possono quindi leggere il capitolo 8 per imparare le funzioni specifiche del Basic realizzate sul **computer**. Poiché ogni dialetto del Basic è alquanto diverso, leggete questo capitolo almeno una volta prima di cominciare a programmare seriamente.

Se non avete mai programmato prima in Basic vi suggeriamo di acquistare un libro sulla programmazione in Basic o di frequentare un corso di Basic prima di cercare di applicarvi allo studio di questi capitoli. Questo manuale non è destinato ad insegnarvi la programmazione.

Il resto del manuale è costituito da:

* Capitolo 7 – informazioni base sull'interfaccia opzionale stampante/registratore a cassette CE-126P.
* Capitolo 9 – è una guida all'individuazione dei guasti per aiutarvi a risolvere alcuni dei problemi di funzionamento e di programmazione.
* Capitolo 10 – Tratta la cura e la manutenzione del vostro nuovo computer.

Delle appendici dettagliate vi danno suggerimenti utili, tabelle e confronti concernenti l'uso ed il funzionamento del **computer**.

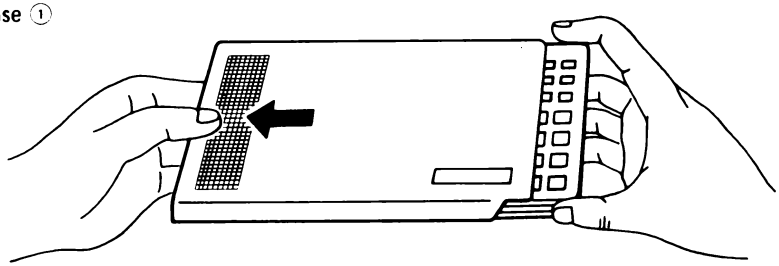
Uso dell'astuccio rigido

Quando non usate il computer, montate l'astuccio rigido sul pannello operativo del computer.

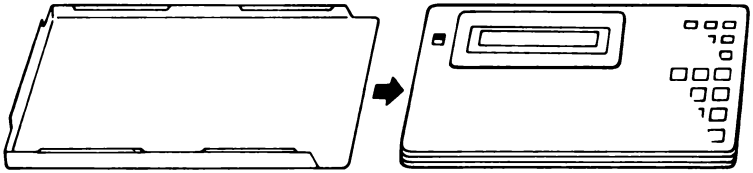
- Quando dovete usare il computer.

Per togliere l'astuccio rigido dal computer, procedete come illustrato nella figura seguente.

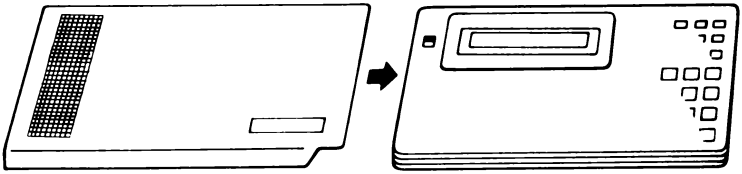
Fase ①



Fase ②



- Quando non usate il computer.

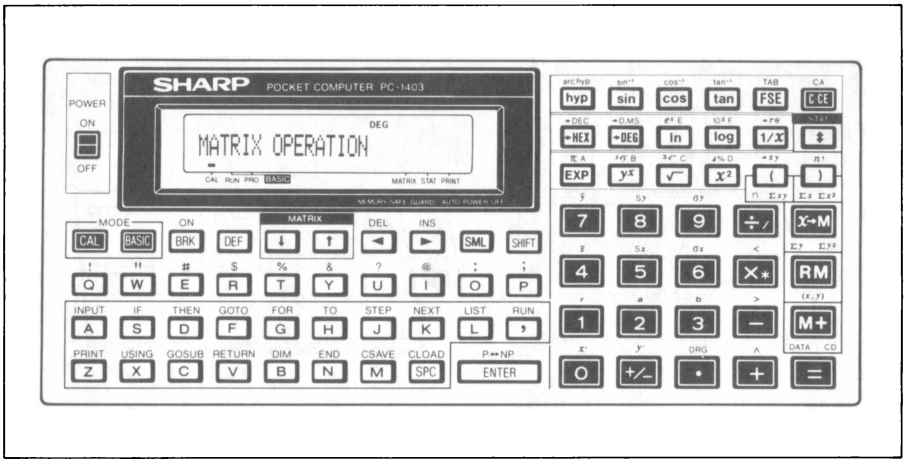


CAPITOLO 2
INTRODUZIONE

Descrizione del sistema

Il Sistema **SHARP computer** è constituito da:

* Una tastiera da 77 caratteri.
* Visore a 24 caratteri
* Un Basic potente in una ROM da 72K Bytes.
* Un processore CMOS da 8 bit.
* Opzionali: Interfaccia stampante/registratore a cassette CE-126P



Per familiarizzarvi con la posizione e la funzione dei comandi del computer, passeremo ora in rassegna le diverse parti della tastiera. Per cominciare, osservate i vari tasti e leggete le iscrizioni su di essi. Cominceremo ad usarli nel capitolo 3.

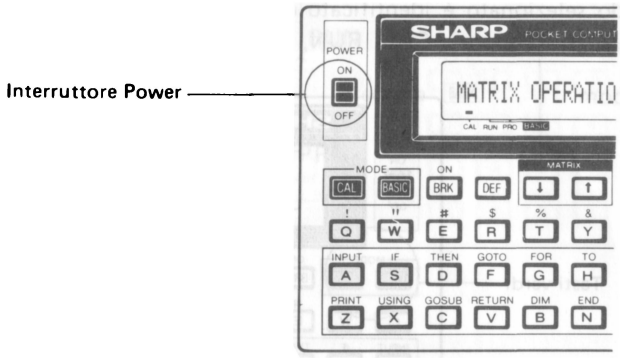
Funzionamento dei tasti e dell'interruttore

Il **computer** ha 77 tasti ed un interruttore a scorrimento. Ogni funzione dei tasti è identificata da diversi caratteri, numeri o simboli indicati sopra o a lato dei tasti.

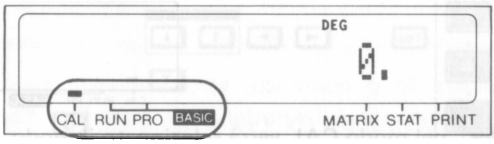
(1) Accensione

Per cominciare, accendete il vostro computer.

Mettete l'interruttore POWER che si trova nell'angolo superiore sinistro del computer in posizione ON.



Sul visore compariranno le seguenti informazioni iniziali:

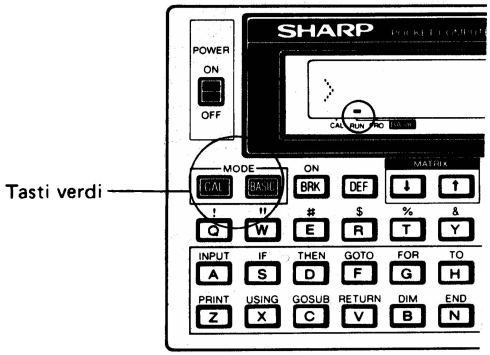


Un indicatore costituito da un (▬) nell'area inferiore sinistra del visore indica il modo in cui è impostato il computer. Quando il computer viene acceso, funziona come una calcolatrice. Per indicare che il computer è impostato nel modo calcolatrice, sopra a CAL (CALcolatrice) compare un trattino.

Per l'uso del computer nel modo CAL, consultate il CAPITOLO 3, "USO COME CALCOLATRICE".

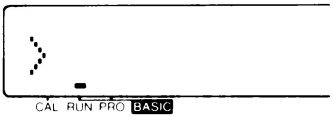
Modi

Il **computer** può funzionare fondamentalmente in tre modi diversi. Un modo è il modo CAL, nel quale usate il vostro computer esattamente come una calcolatrice.
Un altro modo è il modo RUN in cui potete eseguire i programmi o l'elaborazione manuale usando i comandi Basic.
Il terzo modo è il modo PRO che vi consente di memorizzare il programma nel computer o di correggere o modificare un programma memorizzato.
La commutazione tra questi modi può essere effettuata mediante i tasti, [CAL] e [BASIC] . Il modo selezionato è identificato da un indicatore costituito da un trattino (▬) visualizzato sopra a CAL, RUN, o PRO nell'area inferiore sinistra del visore.

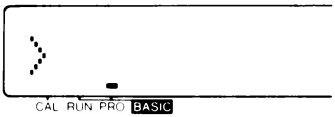


Spegnete ora il computer e riaccendetelo. Verrà selezionato il modo CAL.

Se premete il tasto [BASIC] nel modo CAL verrà selezionato il modo RUN.

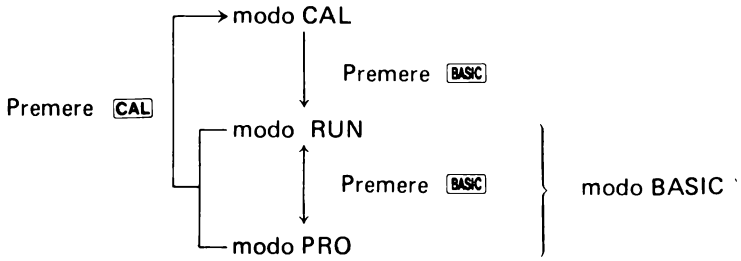


Se premete il tasto [BASIC] nel modo RUN, verrà selezionato il modo PRO.



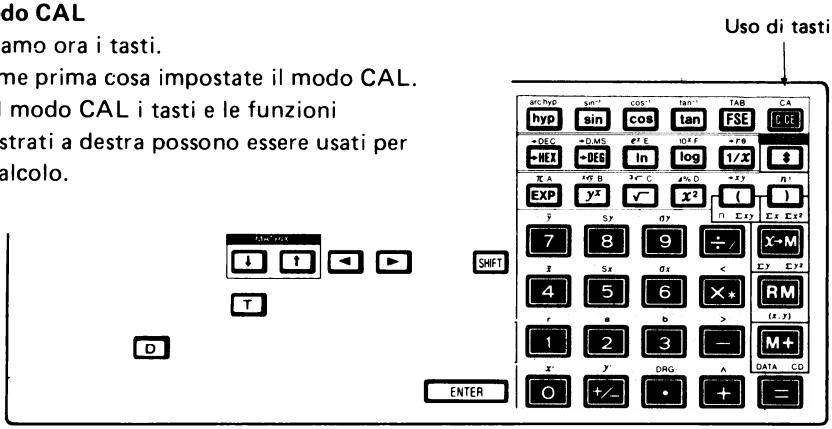
Pertanto i modi RUN e PRO vengono selezionati alternativamente ogni volta che premete il tasto [BASIC] .
Il computer ritorna al modo CAL se premete il tasto [CAL] .

Commutazione di modo



1. Modo CAL

Usiamo ora i tasti.
Come prima cosa impostate il modo CAL.
Nel modo CAL i tasti e le funzioni illustrati a destra possono essere usati per il calcolo.

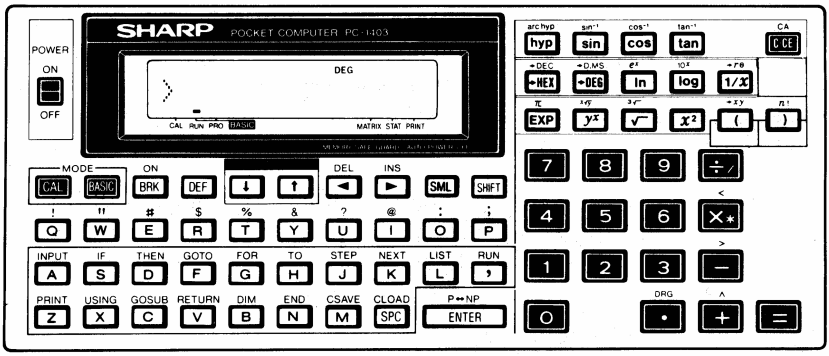


		Visore
[C·CE] (Uso di tasti)	→	0.
[1] [2]	→	12.
[+]	→	12.
[3]	→	3.
[=]	→	15.

2. Modi RUN e PRO

Passate al modo RUN o al modo PRO usando il tasto [BASIC] , e premete i tasti seguenti osservando il visore:

Nei modi RUN e PRO, i tasti illustrati qui di seguito possono essere usati per i calcoli.



Esempio:

[PRINT Z] [USING X] [GOSUB C] → ZXC_

[r 1] [a 2] [DRG •] [b 3] → ZXC12.3_

[CA C·CE] → >

[INPUT A] [=] [x̄ 4] [^ +] [Sx 5] → A = 4 + 5_

↑

Cursore

Se premete un tasto alfabetico o numerico, verrà immesso quanto indicato sul tasto. Quando desiderate immettere il carattere o il simbolo indicato in marrone sopra ciascun tasto, premete il tasto [SHIFT] prima di premere il tasto voluto.

[CA C·CE] [SHIFT] [PRINT Z] → PRINT _

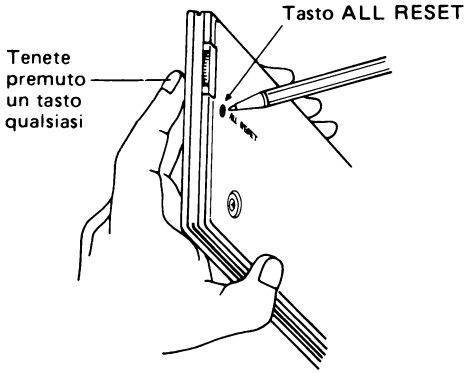
[SHIFT] [" W] [(] [√] → PRINT "(√ _

Il tasto [SHIFT] viene usato per immettere i caratteri o simboli indicati in marrone sopra a ciascun tasto che ha due o tre funzioni. Se premete ripetutamente il tasto [SHIFT] , il simbolo SHIFT nella parte superiore del visore compare e scompare. Il simbolo SHIFT indica che è attivato il tasto [SHIFT] e che è possibile immettere i caratteri indicati in marrone.

Tasto ALL RESET

ALL RESET: Quando il computer si blocca (i tasti non sono operativi) e, non è sufficiente premere [C·CE] o CA ([SHIFT] e [C·CE]) per sbloccarlo, bisogna ricorrere al tasto ALL RESET.

Per azzerare il **computer** tenete premuto un tasto qualsiasi della tastiera e premete simultaneamente il tasto RESET sul retro. Questo preserva tutti i programmi e le variabili.



Nota: Quando premete il tasto All Reset, continuate a premere per almeno 2 o 3 secondi. Lasciando libero il tasto All Reset troppo rapidamente, la sua funzione potrebbe non venire impostata. Premete il tasto All Reset con un qualsiasi oggetto appuntito, come ad esempio una penna a sfera. Non usate oggetti a punta che si spezzano con facilità, come ad esempio matite ed aghi, oppure di diametro superiore a quello al foro del tasto.

Se, nonostante questa operazione, il computer continuasse a non funzionare, premete il solo tasto All Reset e sul visore comparirà il seguente messaggio :

BUSY
MEMORY ALL CLEAR O.K.?

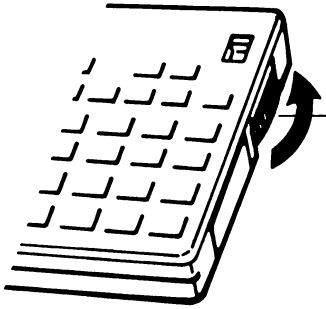
Premete poi il tasto [ENTER] oppure quello [Y] o [=] .

Nota: Se entro i due minuti successivi la comparsa del messaggio sopra riportato nessun tasto viene premuto, il computer si spegne automaticamente.

Dato che quest'operazione cancella ogni contenuto della memoria (dati e programmmi), il tasto RESET deve venire premuto senza premere altri tasti solo in caso di assoluta necessità.

Controll del contrasto

Il computer possiede sul lato destro del pannello frontale visto dal davanti un comando pei la regolazione del contrasto. Regolate con esso la visibilità delle cifre sul visore.



Controllo del contrasto
Girate il comando nella direzione della freccia (antioraria) per aumentare il contrasto, ed in direzione contraria (oraria) per diminuirlo.

Sostituzione delle Batterie

Il calcolatore viene normalmente alimentato da due batterie al litio.

Quando sostituite le pile, l'osservanza delle avvertenze seguenti vi eviterà molti problemi

- Sostitute sempre entramble le pile contemporaneamente.
- Non abbinate una pila nuova ad una usata.
- Usate soltanto pile al litio (tipo CR-2032). Sono necessarie due pile.

Quando sostituire le batterie

Quando il visore diventa evanescente anche girando al massimo in senso antiorario la manopola di regolazione del contrasto sulla destra del computer, significa che le pile sono scariche. In questo caso, sostituitele immediatamente.

Nota: Se usate l'interfaccia a cassette CE-126P ed il registratore a nastro CE-152 opzionali, prima di sostituire le batterie, salvate su nastro i vostri programmi e dati contenuti in memoria.

Come sostituire le batterie

(1) Spegnete il computer portando l'interruttore POWER su OFF.

(2) Estraete le viti dal retro dell'unità con un piccolo cacciavite a croce (Fig. 1).

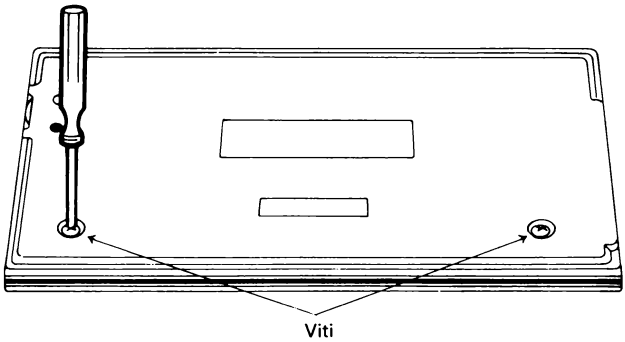


Fig. 1

(3) Togliete il coperchio del vano batterie facendolo scivolare nella direzione indicata dalla freccia in Fig. 2.

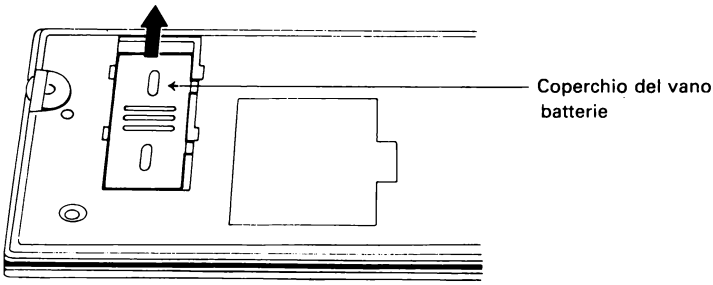


Fig. 2

(4) Sostituite le due batterie (Fig. 3).

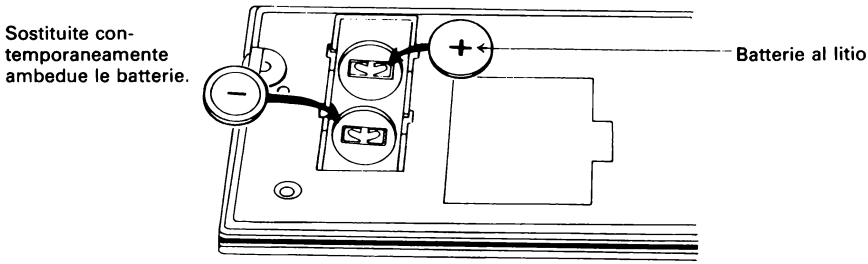


Fig. 3

(5) Rimettete il coperchio del vano batterie facendolo scivolare nella direzione opposta a quella indicata dalla freccia in Fig. 2.

(6) Inserite i cardini del pannello posteriore nelle fessure nel corpo del computer (Fig. 4).

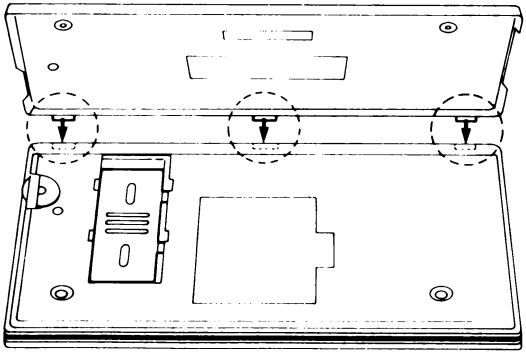
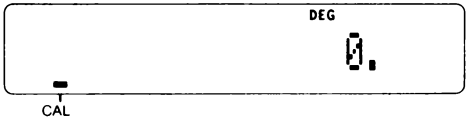


Fig. 4

(7) Mentre rimontate le viti, premete leggermente il pannello posteriore.

(8) Riaccendete il computer portando l'interruttore POWER su ON e premete quindi il tasto RESET per liberarne la memoria. Premete infine [ENTER].
Sul visore dovrebbe apparire quanto segue :



Se il visore è vuoto oppure visualizza indicazioni diverse da "_ 0.", togliete le batterie rimontandole quindi subito dopo, quindi ricontrollate il visore.

Nota: Lasciando una batteria scarica all'interno del computer potreste causare perdite del liquido che essa contiene. Togliete subito le batterie scariche.

AVVERTENZA: Tenete le batterie lontano dalla portata di bambini.

CAPITOLO 3
USO COME CALCOLATRICE

Ora che conoscete l'impostazione ed i componenti del **computer**, cominceremo ad esaminarne le notevoli capacità.

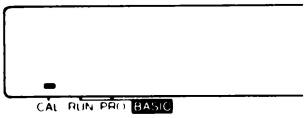
Poiché il **computer** vi offre l'intera gamma delle funzioni di calcolo, oltre all'aumentata potenza delle capacità di programmazioni in Basic (utile nei calcoli più complessi), esso viene comunemente chiamato un computer "intelligente". Ciò naturalmente fa di voi un utente "intelligente"!

(Prima di usare il **computer**, accertatevi che le pile siano correttamente installate).

Accensione

Per accendere il **computer**, fate scorrere l'interruttore verso l'alto in posizione ON.

Quando desiderate usare il vostro **computer** come calcolatrice scientifica, mettete il computer nel modo CAL. Il modo CAL può essere selezionato quando il computer viene acceso o quando viene premuto il tasto [CAL]. Una volta selezionato il modo CAL, compare un trattino (**–**) sopra CAL nell'area inferiore di sinistra del visore.



Se il trattino (–) è in corrispondenza di RUN o di PRO, premete il tasto [CAL] oppure accendete e spegnete di seguito il calcolatore per selezionare il modo CAL.

Spegnimento

Per spegnere il **computer**, fate scorrere l'interruttore di alimentazione nella posizione OFF.

Quando spegnete il computer, cancellate il visore.

Auto Off (Spegnimento automatico)

Per evitare di consumare le pile, il **computer** si spegne automaticamente quando non viene premuto alcun tasto per circa 11 minuti. (Nota: il **computer** non si spegne automaticamente quando state eseguendo un programma).

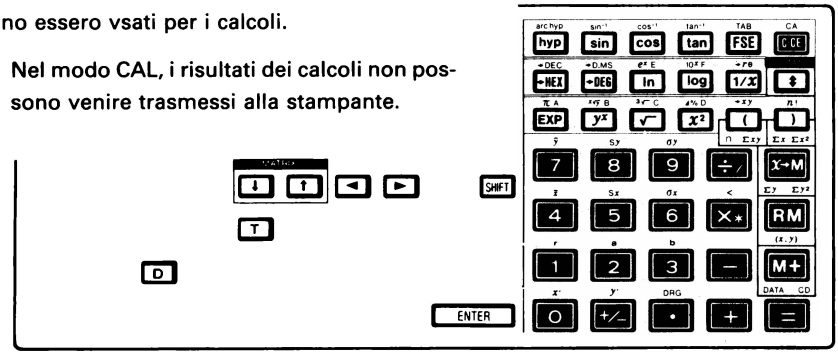
Per riaccendere il computer dopo uno spegnimento automatico, premete il tasto [ON BRK] che si trova alla destra dei tasti verdi.

Notate che il modo CAL può essere selezionato quando viene premuto il tasto ON BRK

Tasti usati per i calcoli nel modo CAL

Nel modo CAL i tasti e le funzioni illustrati a destra, possono essero vsati per i calcoli.

Nota: Nel modo CAL, i risultati dei calcoli non possono venire trasmessi alla stampante.



Facciamo ora alcuni semplici calcoli. Premete i tasti seguenti osservando il visore:

Inserire	Visore
r 1 a 2 b 3	123.
^ +	123.
σx 6 Sx 5 x̄ 4	654.
=	777.

↑

Premete il tasto dell'uguale come in una calcolatrice normale.

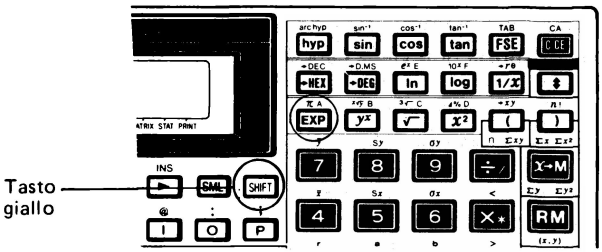
(123 + 654 = 777)

Avete ottenuto il risultato giusto? In caso contrario, spegnete il computer, quindi accendetelo nuovamente e riprovate lo stesso calcolo.

Richiamiamo ora il valore del pi greco (π).

La lettera "π" è riportata in marrone sopra al tasto EXP . Le funzioni identificate dalle lettere marroni possono essere usate premendo dapprima il tasto SHIFT , e quindi il tasto funzionale voluto.

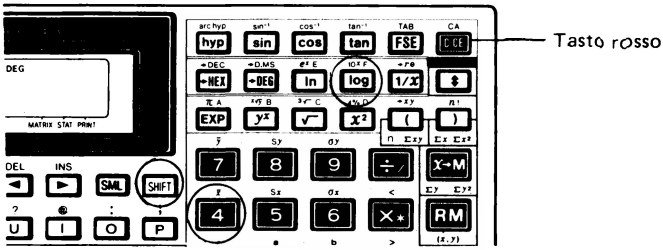
Premete ora [SHIFT] [EXP] .



Inserire	Visore
[SHIFT] [π A / EXP]	3.141592654

($\pi \doteqdot 3.141592654$)

Quello che vedete sul visore è il valore del pi greco.
Calcoliamo quindi 10^4. Per questo calcolo dovete usare la funzione 10^x.
Anche questa funzione è identificata da una lettera marrone; pertanto il tasto [SHIFT] deve essere premuto prima di premere il tasto funzionale:



Inserire	Visore
[4] [SHIFT] [log]	10000.

($10^4 = 10000$)

Quanto segue spiega le funzioni di alcuni tasti.

* [C·CE] (cancellazione) (tasto rosso)

Se questo tasto viene premuto immediatamente dopo che è stato immesso un dato

numerico o dopo che è stato richiamato il contenuto della memoria, questi dati verranno cancellati. In qualsiasi altro caso, l'uso del tasto [C·CE] cancella gli operatori e i dati numerici immessi.
Il contenuto della memoria non viene cancellato con il tasto [C·CE] .

Inserire	Visore
123 [+] 456	456.
[C·CE]	0.
789 [=]	912.
	(123 + 789 = 912)

Inserire	Visore
6 [×] 2 [+]	12.
[C·CE]	0.
6 [÷] 2 [+]	3.
5 [=]	8.

Il tasto [C·CE] può essere usato anche per cancellare un errore.

Inserire	Visore
5 [÷] 0 [=]	0. E ← Simbolo d'errore
[C·CE]	0.

* [FSE] (Commutazione del modo di visualizzazione)

Questo tasto viene usato per commutare il modo di visualizzazione del risultato di un calcolo dal sistema a punto decimale mobile (modo normale) a quello a punto decimale fisso, a quello di notazione scientifica o di notazione da ingegneria e viceversa.

Inserire	Visore	
23 [×] 1000 [=]	23000.	(Modo normale)
[FSE]	FIX 23000.000	(Modo FIX)
[FSE]	SCI 2.300E 04	(Modo SCI)
[FSE]	ENG 23.000E 03	(Modo ENG)

* [TAB] (Specifica il numero di posizioni decimali)

Questo tasto, se impiegato insieme ad un tasto numerico, specifica il numero di posizioni decimali. Accendete e quindi riaccendete subito dopo il computer. Premete il tasto [FSE] ed il visore indicherà "0.000" (modo FIX).

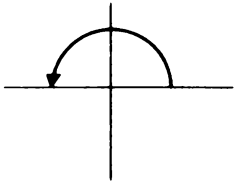
	Inserire	Visore
(1) Specifica 2 posizioni decimali.	[SHIFT] [TAB FSE] [2]	FIX 0.00 (1)
	5 [÷] 8 [=]	FIX 0.63
(2) Specifica 5 posizioni decimali.	[SHIFT] [TAB FSE] [5]	FIX 0.62500 (2)

* [DRG] (specifica i dati angolari).

Questo tasto viene usato per specificare le unità angolari per i dati usati nelle funzioni trigonometriche, nelle funzioni trigonometriche inverse o nella conversione di coordinate.

Inserire	Visore	
	DEG	(Grado sessagesimale)
[SHIFT] [DRG •]	RAD	(Radiante)
[SHIFT] [DRG •]	GRAD	(Grado centesimale)
[SHIFT] [DRG •]	DEG	(Grado sessagesimale)

$180^\circ = \pi\,(\text{rad}) = 200^g$



DEG: Grado sessagesimale [∘]

RAD: Radiante [rad]

GRAD: Grado centesimale [g]

* [0] a [9], [•], [EXP] e [+/−]

[EXP] : Viene usato per immettere i dati con il punto mobile (il visore indica "E" per i dati con il punto mobile).

Inserire	Visore
4 [π A / EXP] 3	4.E 03
	(4×10^3)
[=]	4000.
[+/−]	−4000.

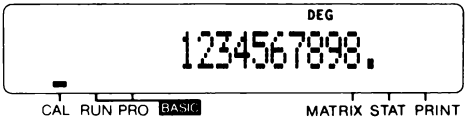
[+/−] : Viene usato per immettere dei dati numerici negativi (oppure per invertire il segno da negativo a positivo).

Inserire	Visore
1.23 [+/−]	−1.23
[EXP] 5 [+/−]	−1.23E−05
	(-1.23×10^{-5})
[=]	−0.0000123
[+/−]	0.0000123

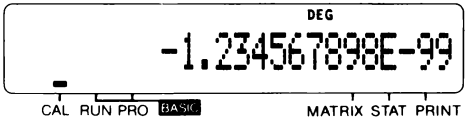
Come si legge il visore

Questa sezione descrive i formati ed i simboli del visore usati nel modo CAL.

Formato normale di visualizzazione



Formato esponenziale di visualizzazione

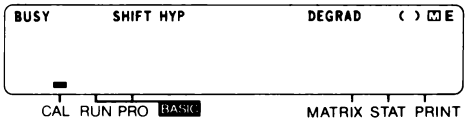


Mantissa (12 cifre) Esponente (4 cifre)

Il computer è dotato di un visore a 24 cifre, delle quali 16 sono usate per visualizzare numeri. Nel modo CAL, i risultati dei calcoli sono normalmente visualizzati nel modo con punto decimale mobile. Se il risultato è inferiore a 0,000000001 oppure superiore a 9999999999 (maggiore di − 0,0000000001 o inferiore a −9999999999), viene visualizzato in formato esponenziale. Nel formato esponenziale, la mantissa di un numero viene visualizzata fino a 12 cifre significative, mentre l'esponente viene visualizzato fino a 4 cifre significative (inclusi il punto decimale, il segno ed il simbolo).

Simboli del visore

Quanto segue descrive i simboli e gli indicatori di stato visualizzati sul visore.



Il modo CAL usa i simboli e gli indicatori sopra illustrati, il cui significato viene descritto qui di seguito:

SHIFT: Questa parola compare quando è attivato il tasto [SHIFT], ad indicare che è possibile selezionare le funzioni indicate dalle etichette marrone chiaro sui tasti. Per abbandonare il modo SHIFT, Premete una seconda volta il tasto [SHIFT].

HYP: Questa parola compare, quando viene premuto il tasto [hyp], ad indicare che è stata selezionata una funzione iperbolica. Se vengono premuti i tasti [SHIFT] [hyp], compare una frase, SHIFT HYP, ad indicare che è stata selezionata una funzione iperbolica inversa.

DEG RAD : GRAD Queste parole vengono selezionate in sequenza ogni volta che vengono premuti i tasti [SHIFT] [DRG •]. Ciascuna di queste parole indica le unità angolari rispettivamente per le funzioni trigonometriche, le funzioni trigonometriche inverse e la conversione delle coordinate.

DEG: grado sessagesimale [∘]

RAD: radiante [rad]

GRAD: grado centesimale [g]

(180 gradi = π rad = 200^g)

( ): Questo simbolo compare quando vengono usate le parentesi in una formula di calcolo mediante il tasto [(].

M : Questo simbolo compare, quando nella memoria di calcolo vengono memorizzati dei dati numerici diversi da zero, ad indicare che la memoria è occupata.

E : Questo simbolo compare se si è verificato un errore. L'errore può essere annullato premendo il tasto [C·CE] .

STAT: Premendo i tasti [SHIFT] [STAT ↕] si attiva un trattino (-) sopra a STAT nell'area inferiore destra del visore. STAT sta per statistica, ed indica che il computer si trova nel modo di calcolo statistico.

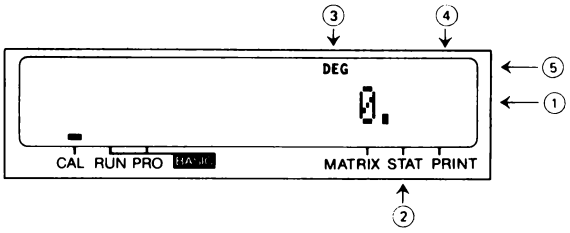
CAL : Se viene visualizzato un trattino (-) sopra a CAL, ciò indica che il computer è nel modo CAL.

BUSY: Questo indicatore compare mentre il computer sta eseguendo una operazione aritmetica.

MATRIX: Premendo [SHIFT] [↑] oppure [SHIFT] [↓] nel modo CAL, il trattino (-) compare sopra MATRIX, sotto il visore in basso. Tale indicatore avverte che il calcolatore è pronto per operazioni su matrici.
Per abbandonare il modo MATRIX, premete di nuovo una delle due combinazioni di tasti.

Calcoli

Questa sezione descrive le operazioni fondamentali nel modo CAL eseguite dal computer.
Prima di cominciare, impostate correttamente il computer.
Come prima cosa, premete il tasto [CAL] per mettere il computer nel modo CAL. Premete quindi [C·CE] [C·CE] , ed accertatevi che il visore visualizzi le seguenti informazioni iniziali.



Se il visore non visualizza queste informazioni, leggete la descrizione seguente e prendete le misure del caso:

① Viene visualizzato più di uno zero (ad es. 0.00):
Il numero di cifre dopo la virgola viene specificato. Cancellate il modo TAB spegnendo e quindi riaccendendo il computer. Il computer si porta così al modo normale del display.

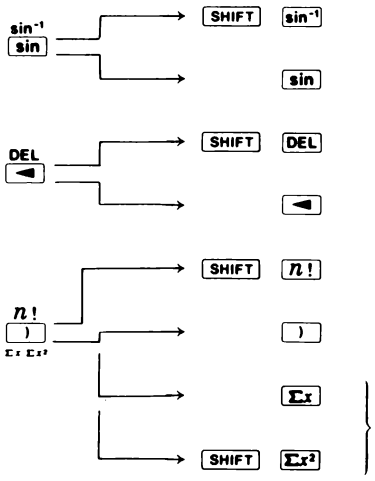
② In corrispondenza delle indicazioni STAT o MATRIX appare un trattino (-) : Il computer si trova nel mode di calcolo. Premete [SHIFT] [STAT] per abbandonare il mode STAT. Premete [SHIFT] [↓] o [SHIFT] [↑] per abbandonare il modo MATRIX.

③ Invece di DEG è visualizzato RAD o GRAD: RAD, GRAD e DEG indicano delle unità angolari per i dati visualizzati. Può essere visualizzato uno qualsiasi di questi simboli, a meno che debba essere eseguita una funzione trigonometrica, una funzione trigonometrica inversa o una conversione di coordinate. E' possibile selezionare in sequenza ciascuno di questi simboli premendo [SHIFT] [DRG] .

④ Viene visualizzato il simbolo [M] : Un dato numerico è già in memoria. E' possibile cancellarlo premendo i tasti [C·CE] [x→M] .

⑤ Tutti i simboli visualizzati nell'area superiore del visore possono essere cancellati con il tasto [C·CE] , con l'eccezione di quelli descritti ai punti ③ e ④ .

In questo manuale, illustreremo sempre le funzioni dei tasti come segue:



tasto $\sin^{-1}$

tasto sin

tasto di cancellazione

tasto della freccia a sinistra

tasto fattoriale

tasto della parentesi chiusa

Questi tasti vengono attivati quando è impostato il modo del calcolo statistico.

1. Addizione-sottrazione

Scrivete quanto segue:

12 [+] 45.6 [−] 32.1 [+] 789 [−] 741 [+] 213 [=]

Risultato: 286.5

2. Moltiplicazione-divisione

a. Scrivete quanto segue:

841 [X] 586 [÷] .12 [=] Risultato: 41068.83333

b. Scrivete quanto segue:

427 [+] 54 [X] 32 [÷] 7 [−] 39 [X] 2 [=] Risultato: 595.8571429

Notate che la moltiplicazione e la divisione vengono risolte prioritariamente rispetto all'addizione ed alla sottrazione. In altre parole, la moltiplicazione e la divisione vengono eseguite prima dell'addizione e della sottrazione.

Moltiplicazione con costante: il primo numero immesso è la costante.

Scrivete: 3 [X] 5 [=] Risultato: 15

Scrivete: 10 [=] Risultato: 30

Divisione con costante: il numero immesso dopo il segno di divisione è la costante.

Scrivete: 15 [÷] 3 [=] Risultato: 5

Scrivete: 30 [=] Risultato: 10

Nota: La macchina mantiene alcuni calcoli a seconda del livello di priorità. Di conseguenza, nel calcolo successivo, l'operatore dell'ultimo calcolo e l'ultimo valore numerico vengono trattati rispettivamente come un'istruzione di calcolo e una costante per il calcolo con costante.

a + b x c =	+bc	(addizione con costante)
a x b ÷ c =	÷ c	(divisione con costante)
a ÷ b x c =	$\frac{a}{b}$ x	(moltiplicazione con costante)
a x b − c =	− c	(sottrazione con costante)

3. Calcoli in memoria

La memoria accessibile indipendentemente è indicata dai tre tasti: [x→M] , [RM] , [M+] . Prima di cominciare un calcolo, azzerate la memoria premendo [C·CE] e [x→M] .

Scrivete: 12 [+] 5 [M+] Risultato: 17

Per sottrarre, scrivete: 2 [+] 5 [=] [+/−] [M+]

Il risultato di questa equazione è: −7

Premete [RM] per richiamare la memoria: 10

Scrivete: 12 [X] 2 [=] [x→M]

Risultato: 24 (sostituisce inoltre il 10 nella memoria)

Scrivete: 8 [÷] 2 [M+]

Risultato: 4 [RM] : 28

Nota:
- I calcoli con la memoria sono impossibili nel modo "STAT" (Modo di calcolo statistico).
- Quando sottraete un numero dalla memoria, premete i tasti [+/−] e [M+] .

Calcoli scientifici nel modo CAL

Per eseguire le funzioni trigonometriche e trigonometriche inverse nonché la conversione delle coordinate, impostate un modo angolare. Il modo angolare viene indicato mediante i tasti [SHIFT] e [DRG] .

1. Funzioni trigonometriche

Impostate il modo angolare "DEG".

Calcolate: Sin 30° + Cos 40° =

Scrivete quanto segue: 30 [sin] + 40 [cos] [=]

Risultato: 1.266044443

Calcolate: Cos 0.25π

Impostate il modo angolare "RAD".

Scrivete: .25 [×] [SHIFT] [π] [=] [cos]

Risultato: 0.707106781

(Ricordate di usare il tasto [SHIFT])

2. Funzioni trigonometriche inverse

Calcolate: Sin^{-1} 0.5

Impostate il modo angolare "DEG".

Scrivete: .5 [SHIFT] [sin⁻¹]

Risultato: 30

Calcolate: Cos^{-1} −1

Impostate il modo angolare "RAD".

Scrivete: 1 [+/−] [SHIFT] [cos⁻¹]

Risultato: 3.141592654

(valore di π)

(Per specificare un numero negativo, premete il tasto [+/−] dopo i numeri).

Il risultato delle funzioni trigonometriche inverse verrà visualizzato nell'area seguente.

$\theta = \sin^{-1} x$, $\theta = \tan^{-1} x$	$\theta = \cos^{-1} x$
DEG: $-90 \leqq \theta \leqq 90$ [°]	DEG: $0 \leqq \theta \leqq 180$ [°]
RAD: $-\frac{\pi}{2} \leqq \theta \leqq \frac{\pi}{2}$ [rad]	RAD: $0 \leqq \theta \leqq \pi$ [rad]
GRAD: $-100 \leqq \theta \leqq 100$ [g]	GRAD: $0 \leqq \theta \leqq 200$ [g]

3. Funzioni iperboliche ed iperboliche inverse

Calcolate: Sinh 4

Scrivete: 4 [hyp] [sin]

Risultato: 27.2899172

Calcolate: Sinh^{-1} 9

Scrivete: 9 [SHIFT] [arc hyp] [sin]

Risultato: 2.893443986

4. Potenze

Calcolate: 20^2

Scrivete: 20 [x^2]

Risultato: 400

Calcolate: 3^3 e 3^4

Scrivete: 3 [y^x] 3 [=]

Risultato: 27

Scrivete: 3 [y^x] 4 [=]

Risultato: 81

5. Radici

Calcolate: $\sqrt{25}$

Scrivete: 25 [$\sqrt{\ }$]

Risultato: 5

Calcolate: radice cubica di 27

Scrivete: 27 [SHIFT] [$\sqrt[3]{\ }$]

Risultato: 3

Calcolate la radice quarta di 81

Scrivete: 81 [SHIFT] [$\sqrt[x]{y}$] 4 [=]

Risultato: 3

6. Funzioni logaritmiche

Calcolate: ln 21, log 173

Logaritmi naturali: Scrivete: 21 [ln]

Risultato: 3.044522438

Logaritmi comuni: Scrivete: 173 [log]

Risultato: 2.238046103

7. Funzioni esponenziali

Calcolate: $e^{3.0445}$

Scrivete: 3.0445 [SHIFT] [e^x]

Risultato: 20.99952881 (21 come al punto "6" precedente)

Calcolate: $10^{2.238}$

Scrivete: 2.238 [SHIFT] [10^x]

Risultato: 172.9816359 (173 come al punto "6" precedente)

8. Reciproci

Calcolate: 1/6 + 1/7

Scrivete: 6 [$1/x$] [+] 7 [$1/x$] [=]

Risultato: 0.309523809

9. Fattoriale

Calcolate: 69!

Scrivete: 69 [SHIFT] [n!]

Risultato: 1.711224524E 98 ($1.711224524 \times 10^{98}$)

Notate che la sezione Errore tratta i limiti di calcolo della calcolatrice.

Calcolate: ${}_8P_3 = \dfrac{8!}{(8-3)!} =$

Scrivete: 8 [SHIFT] [n!] [÷] [(] 8 [−] 3 [)] [SHIFT] [n!] [=]

Risultato: 336

10. Calcoli delle percentuali

Calcolate: 45% di 2,780 ($2{,}780 \times \dfrac{45}{100}$)

Scrivete: 2780 [×] 45 [SHIFT] [Δ%]

Risultato: 1251

Calcolate: $\dfrac{547-473}{473} \times 100$

Scrivete: 547 [−] 473 [SHIFT] [Δ%]

Risultato: 15.6448203

11. Conversioni angolo/tempo

Per convertire un angolo dato in gradi e minuti/secondi nell'equivalente decimale, esso deve essere immesso rispettivamente come numero intero e decimale.

Convertite 12°47'52" nel suo equivalente decimale

Scrivete: 12.4752 [→DEG]

Risultato: 12.79777778

Quando si convertono i gradi decimali nell'equivalente gradi/minuti/secondi, il risultato è scomposto in: parte intera = gradi; prima e seconda cifra decimale = minuti; terza e quarta cifra = secondi, e le cifre decimali dalla quinta all'ultima sono i secondi decimali.

Convertite 24.7256 nel suo equivalente in gradi/minuti/secondi

Scrivete: 24.7256 [SHIFT] [→D.MS]

Risultato: 24.433216 cioé 24°43'32"

Un cavallo ha fatto dei tempi per giro di 2 minuti 25 secondi, 2 minuti 38 secondi e 2 minuti 22 secondi. Qual è il tempo medio della corsa?

Scrivete: .0225 [→DEG] [+] .0238 [→DEG] [+] .0222 [→DEG] [=]

Risultato 1: 0.123611111

Scrivete: [÷] 3 [=]

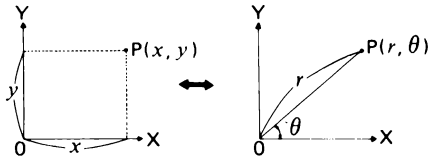
Risultato 2: 0.041203703

Premete: [SHIFT] [→DMS]

Risultato 3: 0.022833333 cioé il tempo medio è di 2 minuti 28 secondi.

12. Conversioni di coordinate

Conversione delle coordinate rettangolari in polari (x, y → r, θ)



$r = \sqrt{x^2 + y^2}$

$\theta = \tan^{-1} \dfrac{y}{x}$

DEG: $0 \leqq |\theta| \leqq 180$

RAD: $0 \leqq |\theta| \leqq \pi$

GRAD: $0 \leqq |\theta| \leqq 200$

Risolvete per $x = 6$ e $y = 4$ Modo = DEG

Scrivete: 6 [↕] 4 [SHIFT] [→rθ]

Risultato: 7.211102551 (r)

Premete: [↕]

Risultato: 33.69006753 (θ)

Calcolate la grandezza e la direzione (fase) in un vettore $\dot{I} = 12 + j9$

Scrivete: 12 [↕] 9 [SHIFT] [→rθ]

Risultato: 15 (r)

Premete: [↕]

Risultato: 36.86989765 (θ)

Conversione delle coordinate polari in rettangolari (r, $\theta \rightarrow x, y$)

Risolvete per P (14, π/3), r = 14, $\theta = \pi/3$

Modo: RAD

Scrivete: [SHIFT] [π] [÷] 3 [=] [↕] 14 [↕] [SHIFT] [→xy]

Risultato: 7.000000002 (x)

Premete: [↕]

Risultato: 12.12435565 (y)

Nell'esempio precedente $\theta = \dfrac{\pi}{3}$ viene immesso per primo e sostituito con r = 14 premendo il tasto [↕] dopo aver immesso r.

Uso delle parentesi

I tasti delle parentesi sono necessari per raggruppare insieme una serie di operazioni quando è necessario escludere il sistema delle priorità algebriche. Quando vengono aperte le parentesi sul visore del **computer** compare il simbolo "( )".
I calcoli tra parentesi sono prioritari rispetto agli altri calcoli. È possibile usare fino a 15 parentesi in unico livello. I calcoli nelle parentesi più interne vengono svolti per primi.

Calcolate: 12 + 42 ÷ (8 – 6)
Scrivete: 12 [+] 42 [÷] [(] 8 [–] 6 [)] [=]
Risultato: 33
Calcolate: 126 ÷ [(3 + 4) x (3 – 1)]
Scrivete: 126 [÷] [(] [(] 3 [+] 4 [)] [×] [(] 3 [–] 1 [)] [)] [=]
Risultato: 9

Nota: E' possibile omettere la chiusura delle parentesi che si trovano prima del [=] o [M+].

Posizione decimali

I tasti [SHIFT] e [TAB] vengono usati per specificare il numero di cifre decimali volute nel risultato. Il numero di posizioni dopo la virgola decimale viene specificato premendo il tasto numerico ([Ø] ~ [9]) dopo i tasti [SHIFT] e [TAB] . In questo caso, il modo del visore deve essere quello FIX (di punto decimale fisso), quello SCI (di notazione scientifica) oppure quello ENG (di notazione da ingegneria)

[SHIFT] [TAB] [Ø] → Indica 0 posizioni decimali
(La prima posizione decimale viene arrotondata)

[SHIFT] [TAB] [1] → Indica una posizione decimale
(La seconda posizione decimale viene arrotondata)

≀

[SHIFT] [TAB] [9] → Indica 9 posizioni decimali
(La decima posizione decimale viene arrotondata)

Per abbandonare la modalità TAB (di designazione delle posizioni decimali) spegnete e riaccendete il computer. Il visore ora si trova nel modo CAL.

Esempio: [SHIFT] [TAB] [9]
[•] [5] [÷] [9] [=] → 0.055555556 (modo FIX)
(La decima posizione decimale viene arrotondata)

[FSE]	→	5.555555556E−02 (modo SCI) (La decima posizione decimale della mantissa viene arrotondata)
[SHIFT] [TAB] [3]	→	5.556E−02 (modo SCI) (La quarta posizione decimale della mantissa viene arrotondata)
[FSE]	→	55.556E−03 (modo ENG)
[FSE]	→	0.055555555 In questo caso la determinazione viene fatta dal computer nella forma di $5.55555555555 \times 10^{-2}$. Arrotondando l'undicesima cifra della mantissa si ottiene $5.555555556 \times 10^{-2}$. Quando si passa al sistema decimale con punto mobile, le parti arrotondate possono non essere visualizzate come in questo esempio.

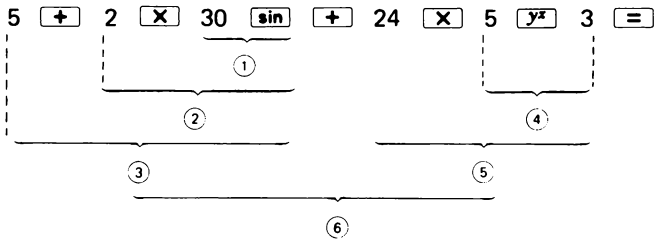
Livelli di priorità

La macchina, essendo dotata di una funzione che giudica il livello di priorità dei singoli calcoli, consente l'azionamento dei tasti secondo una data formula matematica. Quanto segue illustra il livello di priorità dei singoli calcoli.

Livello	**Operazioni**
(1)	Funzioni, quali sin, x^2
(2)	y^x, $\sqrt[x]{y}$
(3)	x, ÷ (I calcoli con lo stesso livello di priorità vengono eseguiti in sequenza)
(4)	+, −
(5)	=, M+, Δ%

Esempio:

Azionamento dei tasti e sequenza di calcolo in $5 + 2 \times \sin 30 + 24 \times 5^3 =$



I numeri ① ~ ⑥ indicano la sequenza in cui vengono svolti i calcoli.

Quando i calcoli vengono svolti da una priorità 1 superiore in sequenza, è necessario riservare una priorità 1 inferiore. La macchina è dotata di memorie di 8 livelli per soddisfare questi requisiti.

Poiché le memorie possono essere usate anche in un calcolo che include delle parentesi, è possibile svolgere un calcolo secondo una data formula matematica a meno che le parentesi e le operazioni in sospeso superino 8 livelli in totale.

- Le funzioni ad una sola variabile vengono calcolate immediatamente dopo l'azionamento del tasto senza essere conservate. (x^2, $1/x$, n!, →DEG, →DMS, etc.)

〈 Calcolo senza l'uso delle parentesi 〉

Esempio:

1 [+] 2 [=] — 1 livello in sospeso

①

1 [+] 2 [×] 3 [=] — 2 livelli in sospeso

① ②

1 [+] 2 [×] 3 [y^x] 4 [=] — 3 livelli in sospeso

① ② ③

1 [+] 2 [×] 3 [y^x] 4 [÷] 5 — Premendo il tasto [y^x], rimangono in sospeso 3 calcoli. Premendo il tasto [÷], si esegue il calcolo di "y^x" che ha il livello di priorità più alto e di "x" che ha un livello di priorità identico. Dopo aver premuto il tasto [÷], rimangono in sospeso gli altri due calcoli.

① ② ③

②

〈 Calcolo con le parentesi 〉

Esempio:

i) 1 [+] 2 [×] 3 [y^x] [(] 4 [÷] 5 — 4 numeri ed istruzioni di calcolo vengono lasciati in sospeso.

① ② ③ ④

ii) 1 [+] 2 [×] [(] 3 [−] 4 [÷] 5 [)] — Premendo il tasto [)] si esegue il calcolo di 3 − 4 ÷ 5 tra parentesi, lasciando in sospeso gli altri due calcoli.

① ② ③ ④

- E' possibile usare le parentesi a meno che i calcoli in sospeso siano più di 8. Tuttavia le parentesi possono essere usate continuamente sino ad un massimo di 15.

  Esempio: a x (((b – c x (((d + e) x f) ÷ g ·····

  Se si continua, le parentesi possono essere usate un massimo di 15.

Conversione tra numeri decimali e numeri esadecimali e calcolo esadecimale ([→HEX] , [→DEC])

[→HEX] : Vi consente di convertire un numero decimale in un numero esadecimale e, contemporaneamente, mettere il **computer** nel modo HEX (il visore visualizza il simbolo HEX).

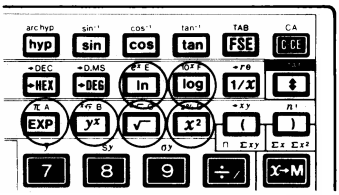
[SHIFT] [→DEC] : Vi consente di convertire un numero esadecimale in un numero decimale ed annulla il modo HEX (il simbolo HEX scompare dal visore).

La notazione esadecimale è uno dei sistemi ampiamente usati nel campo degli elaboratori. La radice della notazione esadecimale è costituita dai numeri 0 ~ 9 e dalle lettere maiuscole A ~ F che vengono usate invece del 10 ~ 15 nella notazione decimale.

(Esadecimale)		(Decimale)
A	———	10
∫		∫
F	———	15

I numeri esadecimali A ~ F possono essere imputati mettendo come prima cosa il computer nel modo Hex (con il tasto [→HEX]), e premendo quindi i rispettivi tasti illustrati nella figura.

Il simbolo HEX indica che i dati numerici visualizzati sul visore sono numeri esadecimali, e che potete eseguire qualsiasi operazione aritmetica sui numeri esadecimali.



Per annullare il modo esadecimale, premete [SHIFT] [→DEC] . Non è possibile annullare questo modo con il tasto [C·CE] .

Conversione da decimale ad esadecimale

(Esempio) Convertite il numero decimale 30 in un numero esadecimale:

Scrivete: 30 [→HEX] Risultato: 1E. HEX

Per eseguire un'altra conversione annullate temporaneamente il mode esadecimale premendo [SHIFT] [→DEC] .

(Esempio) Convertite il numero decimale –2 in un numero esadecimale:
Annullate temporaneamente il modo esadecimale con [C·CE] [SHIFT] [→DEC] .

Scrivete: [2] [+/–] [→HEX]

Risultato: FFFFFFFFFE. HEX

- Se tentate la conversione da decimale in esadecimale su di un numero decimale negativo, il computer esegue internamente il calcolo del "complemento di 2" e visualizza il risultato nel complemento di 16.
- Il tasto [+/–] può essere usato per invertire il segno dei dati numerici presenti sul visore. Se viene invertito il segno di un numero esadecimale positivo, sul visore si ottiene il suo complemento.

(Esempio) Convertite il numero decimale 123.4 in un numero esadecimale.

Scrivete: [SHIFT] [→DEC]
123.4 [→HEX]

Risultato: 7B. HEX

- Se viene convertito in un numero esadecimale un numero decimale avente una parte frazionaria, la parte frazionaria viene arrotondata e soltanto la parte intera viene convertita in un numero esadecimale.

Conversione da esadecimale in decimale

(Esempio) Convertite il numero esadecimale 2BC in decimale.

Scrivete: [C·CE] [→HEX] 2 B C [SHIFT] [→DEC]

Risultato: 700.

(Esempio) Convertite il numero esadecimale FFFFFFFF12 in decimale:

Scrivete: [C·CE] [•HEX] FFFFFFFF 12 [SHIFT] [•DEC]

Risultato: FFFFFFFF12. HEX

−238.

- Se un numero esadecimale compreso tra FFFFFFFFFF e FDABF41C01 viene convertito in decimale, il risultato sarà un numero decimale negativo.

Calcolo esadecimale

Il calcolo esadecimale può essere eseguito dopo che il computer è stato messo nel modo Hex. Premete [C·CE] [•HEX], viene visualizzato il simbolo HEX.

(Esempio) A4 + BA =

Scrivete: A4 [+] B A [=]

Risultato: 15 E. HEX

(350 in decimale)

(Esempio) 8 x 3 =

Scrivete: 8 [×] 3 [=]

Risultato: 18. HEX

(24 in decimale)

(Esempio) (12 + D) x B =

Scrivete: [C·CE] [(] 12 [+] D [)] [×] B [=]

Risultato: 155. HEX

(341 in decimale)

(Esempio) 43A – 3CB =
+) A38 – 2FB =

(Totale)

Scrivete: [C·CE] [x⇔M]

4 3 A [−] 3 C B [M+]

Risultato: 6F. HEX

A 3 8 [−] 2 F B [M+]

Risultato: 73D. HEX

[RM]

Risultato: 7AC. HEX

Per il calcolo esadecimale, dovete notare i punti seguenti:

- Il calcolo esadecimale ignora tutte le parti frazionarie. Ciò significa che il tasto del punto decimale, [•] , non ha alcun significato per il calcolo esadecimale.
- Se un risultato intermedio di un calcolo esadecimale include una parte frazionaria, si incorre in un errore.

  (Esempio) B [÷] 3 [×] . . Errore (viene visualizzato il simbolo "E")

  Se nel risultato finale vi è una parte frazionaria, questa verrà troncata e verrà visualizzata soltanto la parte intera.

  (Esempio) B [÷] 3 [=] . . . 3. HEX
- Nel modo esadecimale, il tasto [+/−] può essere usato per ottenere un complemento del numero esadecimale attualmente visualizzato sul visore.

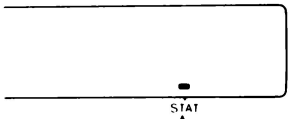
  (Esempio) A B [+/−] → FFFFFFFF55. HEX

  [+/−] → AB. HEX
- Nel modo esadecimale, i tasti funzionali del computer non sono utilizzabili.
- **Quando il computer si trova nei modi STAT o MATRIX (un trattino compare accanto alle indicazioni STAT o MATRIX), la conversione dei numeri da decimali in esagesimali ed il calcolo esagesimale non sono eseguibili.**

Calcolo statistico

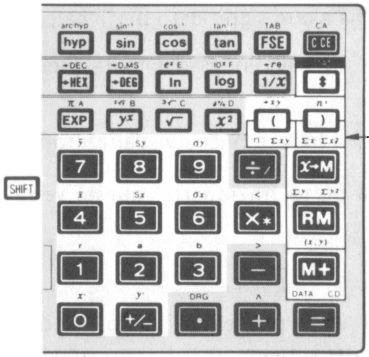
Per eseguire il calcolo statistico, premete i tasti [SHIFT] e [STAT] (sotto il tasto rosso [C·CE]), comparirà un trattino (–) sopra a STAT nell'area inferiore di destra del visore. "STAT" sta per STATistica, ed indica che il computer è nel modo di calcolo statistico.

Quando il computer è in modo RUN o PRO, premete [CAL] e [SHIFT] [STAT] .



Premendo i tasti [SHIFT] e [STAT] , viene visualizzato un trattino in questa posizione.

Tasti che vengono usati essenzialmente nel modo di calcolo statistico.



Quando viene eseguito il calcolo statistico, i dati statistici vengono memorizzati automaticamente nelle variabili fisse (memorie) usate nel modo BASIC. Questi dati statistici possono essere usati nel modo BASIC in quanto vengono conservati in memoria anche quando il modo di calcolo statistico viene annullato.
Questi dati statistici sono cancellati quando il modo statistico viene annullato e immediatamente ripristinato.

Memoria	Z	Y	X	W	V	U
Contenuto	n	Σx	Σx^2	Σxy	Σy	Σy^2

Per cancellare i dati statistici inseriti precedentemente ed i calcoli, annullate il modo statistico e reimpostatelo nuovamente. Diversamente, quando viene eseguito un nuovo calcolo statistico, si otterrà un risultato errato.

Quando è impostato il modo del calcolo statistico, non è possibile eseguire quanto segue:

* Calcolo con la memoria
* Calcolo con le parentesi
* Conversione di coordinate
* Conversioni dalla notazione esadecimale nella notazione decimale e viceversa
* Calcolo esadecimale

1. Calcolo statistico con una variabile

Sapendo che:

(1) n: Numero dei campioni
(2) Σx: Totale dei campioni
(3) Σx^2: Somma dei quadrati dei campioni
(4) $\bar{x}$: Valore medio dei campioni $\bar{x} = \frac{\Sigma x}{n}$
(5) Sx: Deviazione standard con il parametro di popolazione "n–1".

$$Sx = \sqrt{\frac{\Sigma x^2 - n\bar{x}^2}{n-1}}$$

(Usato per valutare la deviazione standard della popolazione dai dati campione estratti da tale popolazione).

(6) σx: Deviazione standard con il parametro di popolazione "n".

$$\sigma x = \sqrt{\frac{\Sigma x^2 - n\bar{x}^2}{n}}$$

(Usato quando tutte le popolazioni vengono assunte come dati campione o quando si cerca la deviazione standard della popolazione con il campione assunto come una popolazione).

I dati per i calcoli statistici ad una sola variabile vengono immessi mediante le operazioni seguenti:

(1) Dati [DATA] (usato per inserire i dati uno ad uno)

(2) Dati [X] Frequenza [DATA] (quando vengono immessi due o più degli stessi dati)

Esempio:

Calcolate la deviazione standard, la media, e la varianza $(Sx)^2$ dai dati seguenti:

Ponete il computer nel modo di calcolo statistico.

Valore	35	45	55	65
Frequenza	1	1	5	2

Di mano in mano che viene immesso ciascun campione, il numero di tale campione compare sul lato destro del visore.

Premere:	Visore:
[SHIFT] [STAT]	0.
35 [DATA]	1.
45 [DATA]	2.
55 x 5 [DATA]	7.
65 x 2 [DATA]	9.

Note: 1. Dopo che tutti i dati sono stati inseriti, dati statistici come il valore della media, la deviazione standard ecc. possono essere ottenuti nell'ordine desiderato.

2. Dopo che il valore della media, della deviazione standard o di un qualsiasi dato statistico è stato ottenuto come risultato intermedio, è possibile introdurre nuovi dati ed eseguire nuovi calcoli in modo continuo su tali nuovi dati.

	Premere:	Visore:
Media:	[SHIFT] [$\bar{x}$]	53.88888889
Deviazione standard:	[SHIFT] [S_x]	9.279607271
Varianza:	[x^2]	86.11111111

Correzione dei dati (CD): l'ultimo dato inserito nell'esempio precedente è errato e deve essere cambiato in 60 x 2.

Premere:	Visore:
65 [X] 2 [SHIFT] [CD]	7.
60 [X] 2 [DATA]	9.

2. Statistica a due variabili e regressione lineare

Oltre alle stesse funzioni statistiche per y come per x nella statistica a variabile singola, viene aggiunta la somma dei prodotti dei campioni Σxy nella statistica a due variabili. La statistica a due variabili rende possibile lo sviluppo di una relazione (correlazione) tra due insiemi di dati. Ogni coppia di dati ha un valore x ed y. Sulla base di questi insiemi di dati, è possibile definire una linea di regressione. La relazione dei due insiemi di dati mediante l'uso del metodo della linea retta viene chiamata "regressione lineare". Nella regressione lineare vi sono tre valori importanti: r, a e b.

L'equazione della linea retta è y = a + bx, dove a è il punto in cui la retta interseca l'asse delle Y e b è l'inclinazione della retta.

Il coefficiente di correlazione r indica la relazione tra due insiemi di dati. La correlazione perfetta tra due valori è r uguale ad 1 (–1 è una correlazione negativa perfetta); in altre parole, sapendo il valore di una variabile, potete prevedere con una precisione del 100% il valore dell'altra variabile. Più il valore di r si allontana da 1, e meno affidabili saranno le vostre previsioni. La tabella seguente può essere usata come un insieme di definizioni dei valori del coefficiente di correlazione:

	Valore di *r*	**Definizione**
Correlazione positiva	+0.80 ~ +1.00	Molto alta
	+0.60 ~ +0.80	Alta
	+0.40 ~ +0.60	Moderata
	+0.20 ~ +0.40	Bassa
	–0.20 ~ +0.20	Zero
Correlazione negativa	–0.20 ~ –0.40	Bassa
	–0.40 ~ –0.60	Moderata
	–0.60 ~ –0.80	Alta
	–0.80 ~ –1.00	Molto alta

r Coefficiente di correlazione

$$r = \frac{S_{xy}}{\sqrt{S_{xx} \cdot S_{yy}}}$$

a $a = \bar{y} - b\bar{x}$

b $b = \dfrac{S_{xy}}{S_{xx}}$

Coefficiente dell'equazione di regressione lineare $y = a + bx$

$$\left[\begin{array}{l} S_{xx} = \Sigma x^2 - \dfrac{(\Sigma x)^2}{n} \\ S_{yy} = \Sigma y^2 - \dfrac{(\Sigma y)^2}{n} \\ S_{xy} = \Sigma xy - \dfrac{\Sigma x \cdot \Sigma y}{n} \end{array}\right]$$

Esempio (1): Se sappiamo il voto in matematica di uno studente, possiamo prevederne il voto in inglese?

Nella tabella seguente vengono dati i voti di esame di 5 studenti scelti casualmente:

Studente N° n	Voto in matematica x	Voto in inglese y
1	82	79
2	53	50
3	61	87
4	74	96
5	51	73
6	51	73

Premere:	Visore
82 [(x.y)] 79 [DATA]	1.
53 [(x.y)] 50 [DATA]	2.
61 [(x.y)] 87 [DATA]	3.
74 [(x.y)] 96 [DATA]	4.
51 [(x.y)] 73 [×] 2 [DATA]	6. (Nota: per inserire campioni identici procedere come indicato)
[SHIFT] [r]	0.571587901
[SHIFT] [a]	34.26190476
[SHIFT] [b]	0.678571428

Il valore di r .57 indica che la correlazione è moderata. L'equazione della linea retta di questi dati è $y = 34.26 + 0.68x$.

Premere:	Visore
90 [SHIFT] [y']	95.33333333

Se avessimo avuto uno studente con un voto in matematica di 90, sulla base di questa analisi, lo studente avrebbe avuto un voto in inglese di 95.

Esempio ② : Il peso è un buon parametro per prevedere la longevità tra gli uomini di 65 anni? Nel 1950, 10 uomini, tutti alti un metro e ottanta, sono stati scelti per un esperimento il cui intento era di determinare se il loro peso avrebbe influenzato la loro durata di vita.

Campione	1	2	3	4	5	6	7	8	9	10
Età alla morte	72	67	69	85	91	68	77	74	70	82
Peso all'età di 65	83	101	90	76	77	88	79	78	89	77

Premere:	Visore
[SHIFT] [STAT]	0.
72 [(x,y)] 83 [DATA]	1.
67 [(x,y)] 101 [DATA]	2.
(Continuate ad immettere tutti i dati)	⋮
[SHIFT] [r]	–0.789763809

r indica una correlazione negativa relativamente alta. Il peso più alto indica una durata di vita più breve. Si usano a e b per tracciare la linea di regressione.

[SHIFT] [a]	143.3817104
[SHIFT] [b]	–0.789161727

Prevedete l'età alla morte di un uomo alto un metro e ottanta che nel 1950 pesava 86 chilogrammi.

86 [SHIFT] [x'] 72.7 anni

Per arrivare all'età di 90 anni, quale dovrebbe essere il peso di un uomo nel 1960?

90 [SHIFT] [y'] 72.36 chilogrammi

Per arrivare all'età di 150 anni, quale dovrebbe essere il peso di un uomo? Ovviamente la risposta non avrebbe alcun significato, il che sta ad indicare il pericolo di portare troppo avanti un'estrapolazione in linea retta.

ATTENZIONE

Dei dati statistici ottenuti nel modo CAL, I SEGUENTI DATI sono conservati nelle locazioni di memoria (da U a Z) utilizzate in BASIC.

Memoria	Z	Y	X	W	V	U
Contenuto	n	Σx	Σx^2	Σxy	Σy	Σy^2

Quando desiderate effettuare dei calcoli utilizzando i dati statistici, usate il modo RUN.

Ad esempio, per determinare la somma dei quadrati (S^2) di quattro dati, 205, 221, 226 e 220, agite sul vostro computer come segue:

$$\begin{aligned} S^2 &= \Sigma(x-\bar{x})^2 \\ &= \Sigma x^2 - n\bar{x}^2 \\ &= \Sigma x^2 - \frac{1}{n}(\Sigma x)^2 \end{aligned}$$

- Immettete i dati nel modo CAL.

[CAL] [SHIFT] [STAT] — Ø.

205 [DATA] 221 [DATA]

226 [DATA] 220 [DATA] — 4.

- Cambiate ora il modo da CAL a RUN e calcolate S^2.

[BASIC] — >

[X] [−] [Y] [X*] [Y] [÷/] [Z] — X−Y * Y/Z_

[ENTER] — 246.

Gamma di calcolo

Quattro calcoli aritmetici:

Primo operando, secondo operando e
risultato del calcolo: $\pm 1 \times 10^{-99} \sim \pm 9.999999999 \times 10^{99}$ e 0

Funzioni scientifiche:

Funzioni	Gamma dinamica	Nota
$\sin x$ $\cos x$ $\tan x$	DEG: $\lvert x \rvert < 1 \times 10^{10}$ RAD: $\lvert x \rvert < \frac{\pi}{180} \times 10^{10}$ GRAD: $\lvert x \rvert < \frac{10}{9} \times 10^{10}$ In $\tan x$, tuttavia, sono esclusi i casi seguenti. DEG: $\lvert x \rvert = 90\,(2n-1)$ RAD: $\lvert x \rvert = \frac{\pi}{2}(2n-1)$ n = intero GRAD: $\lvert x \rvert = 100\,(2n-1)$	
$\sin^{-1} x$ $\cos^{-1} x$	$-1 \leqq x \leqq 1$	
$\tan^{-1} x$	$\lvert x \rvert < 1 \times 10^{100}$	
$\ln x$ $\log x$	$1 \times 10^{-99} \leqq x < 1 \times 10^{100}$	($\ln x = \log_e x$)
e^x	$-1 \times 10^{100} < x \leqq 230.2585092$	($e \fallingdotseq 2.718281828$)
10^x	$-1 \times 10^{100} < x < 100$	
y^x (^)	• $y > 0$: $-1 \times 10^{100} < x \log y < 100$ • $y = 0$: $x > 0$ • $y < 0$: x: n. intero o $\frac{1}{x}$: num. dispari $-1 \times 10^{100} < x \log \lvert y \rvert < 100$	$y^x = 10^{x \cdot \log y}$
$\sqrt[x]{y}$	• $y > 0$: $-1 \times 10^{100} < \frac{1}{x} \log y < 100$, $x \neq 0$ • $y = 0$: $x > 0$ • $y < 0$: x o $\frac{1}{x}$: intero ($x \neq 0$) $-1 \times 10^{100} < \frac{1}{x} \log \lvert y \rvert < 100$	$\sqrt[x]{y} = 10^{\frac{1}{x} \cdot \log y}$
$\sqrt[3]{x}$	$\lvert x \rvert < 1 \times 10^{100}$	
$\sinh x$ $\cosh x$ $\tanh x$	$-227.9559242 \leqq x \leqq 230.2585092$	
$\sinh^{-1} x$	$\lvert x \rvert < 1 \times 10^{50}$	
$\cosh^{-1} x$	$1 \leqq x < 1 \times 10^{50}$	
$\tanh^{-1} x$	$\lvert x \rvert < 1$	

Funzioni		Gamma dinamica	Nota
$\sqrt{x}$		$0 \leqq x < 1 \times 10^{100}$	
x^2		$\lvert x \rvert < 1 \times 10^{50}$	
$\frac{1}{x}$		$\lvert x \rvert < 1 \times 10^{100}$ $x \neq 0$	
n!		$0 \leqq n \leqq 69$ (n: intero)	
D.MS→DEG		$\lvert x \rvert < 1 \times 10^{100}$	
DEG→D.MS		$\lvert x \rvert < 1 \times 10^{100}$	
HEX → DEC		$0 \leqq x \leqq$ 2540BE3FF FDABF41C01 $\leqq x \leqq$ FFFFFFFFFF	x è un numero intero nel modo HEX
DEC → HEX		$\lvert x \rvert \leqq$ 9999999999	x è un intero
$x, y \rightarrow r, \theta$		$(x^2 + y^2) < 1 \times 10^{100}$ $\frac{y}{x} < 1 \times 10^{100}$	$r = \sqrt{x^2 + y^2}$ $\theta = \tan^{-1} \frac{y}{x}$
$r, \theta \rightarrow x, y$		$r < 1 \times 10^{100}$, $\lvert r \sin\theta \rvert < 1 \times 10^{100}$ $\lvert r \cos\theta \rvert < 1 \times 10^{100}$	$x = r \cos\theta$ $y = r \sin\theta$ θ è nella stessa condizione di x, di $\sin x$, $\cos x$.
Calcolo statistico	Data CD	$\lvert x \rvert < 1 \times 10^{50}$ $\lvert y \rvert < 1 \times 10^{50}$ $\lvert \Sigma x \rvert < 1 \times 10^{100}$ $\Sigma x^2 < 1 \times 10^{100}$ $\lvert \Sigma y \rvert < 1 \times 10^{100}$ $\Sigma y^2 < 1 \times 10^{100}$ $\lvert \Sigma xy \rvert < 1 \times 10^{100}$ $\lvert n \rvert < 1 \times 10^{100}$	
	$\bar{x}$	$n \neq 0$	
	Sx	$n \neq 1$ $0 \leqq \frac{\Sigma x^2 - n\bar{x}^2}{n-1} < 1 \times 10^{100}$	
	σx	$n \neq 0$ $0 \leqq \frac{\Sigma x^2 - n\bar{x}^2}{n} < 1 \times 10^{100}$	
	$\bar{y}$	$n \neq 0$	
	Sy	$n \neq 1$ $0 \leqq \frac{\Sigma y^2 - n\bar{y}^2}{n-1} < 1 \times 10^{100}$	
	σy	$n \neq 0$ $0 \leqq \frac{\Sigma y^2 - n\bar{y}^2}{n} < 1 \times 10^{100}$	

Funzioni		Gamma dinamica	Nota
Calcolo statistico	r	$n \neq 0$ $0 < \lvert (\Sigma x^2 - n\bar{x}^2) \cdot (\Sigma y^2 - n\bar{y}^2) \rvert < 1 \times 10^{100}$ $\left\lvert \Sigma xy - \frac{\Sigma x \cdot \Sigma y}{n} \right\rvert < 1 \times 10^{100}$ $\left\lvert \frac{\Sigma xy - \frac{\Sigma x \cdot \Sigma y}{n}}{\sqrt{(\Sigma x^2 - n\bar{x}^2) \cdot (\Sigma y^2 - n\bar{y}^2)}} \right\rvert < 1 \times 10^{100}$	
	b	$n \neq 0$ $0 < \lvert \Sigma x^2 - n\bar{x}^2 \rvert < 1 \times 10^{100}$ $\left\lvert \Sigma xy - \frac{\Sigma x \cdot \Sigma y}{n} \right\rvert < 1 \times 10^{100}$ $\left\lvert \frac{\Sigma xy - \frac{\Sigma x \cdot \Sigma y}{n}}{\Sigma x^2 - n\bar{x}^2} \right\rvert < 1 \times 10^{100}$	
	a	a è la stessa condizione di b, e $\lvert \bar{y} - b\bar{x} \rvert < 1 \times 10^{100}$	
	y'	$\lvert a + bx \rvert < 1 \times 10^{100}$	
	x'	$\left\lvert \frac{y - a}{b} \right\rvert < 1 \times 10^{100}$	

Per quanto concerne la precisione delle funzioni diverse da quelle precedentemente mostrate, l'errore è di norma di ±1 alla decima cifra. (Nel sistema di notazione scientifica, l'errore è di ±1 per la cifra inferiore della mantissa). Tuttavia, la precisione diventa bassa attorno ai singoli punti ed ai punti di inflessione delle funzioni.

Pertanto, gli errori si accumulano ad ogni stadio dei calcoli continui, provocando uno scadimento della precisione. (Altrettanto vale per gli altri calcoli continui eseguiti dalla calcolatrice quali y^x e $\sqrt[x]{y}$.)

Funzione di calcolo matriciale

Nel modo CAL il computer è in grado di calcolare matrici o i loro valori determinanti.

Una matrice è una distribuzione rettangolare a_{ik} (i = 1,2,.... n) di un gruppo dato di numeri (m × n elementi) del tipo mostrato più in basso.

$$\begin{bmatrix} a_{11} & a_{12} \cdots\cdots a_{1n} \\ a_{21} & a_{22} \cdots\cdots a_{2n} \\ \vdots & \vdots \\ a_{m1} & a_{m2} \cdots\cdots a_{mn} \end{bmatrix}$$

Con questo computer, una distribuzione di questo tipo viene indicata come matrice X, Y, o M. Uno dei gruppi che formano la matrice viene chiamato elemento della matrice. L'elemento di matrice a_{11} viene espresso come $X(1,1)$, $Y(1,1)$ oppure $M(1,1)$.
Una disposizione orizzontale di elementi di matrice viene chiamata una riga, mentre una verticale viene chiamata colonna.

Configurazione di matrice
Il computer è in grado di definire tre matrici X,Y e M. Ogni matrice può essere definita in una gamma da 1 a 99 sia orizzontalmente (cioè in righe) che verticalmente (cioè in colonne). Tuttavia, le dimensioni complessive della matrice ricevono i propri limiti dalla capacità di memoria del computer.

Inoltre, le matrici X, Y ed M sono memorizzate nella stessa area di memoria insieme alle distribuzioni BASIC X(*,*),Y(*,*) ed M(*,*). In altre parole, i valori degli elementi di matrice inseriti nel modo BASIC possono essere calcolati nel modo CAL.

Nell'inserire i valori degli elementi di matrice nel modo BASIC, prestate attenzione ai seguenti punti :
1. Gli elementi della matrice $X(1,2)$ corrispondono agli elementi della distribuzione BASIC X(i−1, k−1). Per esempio, $X(1,2)$ corrisponde alla distribuzione X(0,1).
2. Tutti i valori degli elementi di matrice presenti in memoria vengono cancellati dai comandi BASIC RUN, CLEAR e NEW.

Inserimento di elementi di matrice
Il modo MATRIX può venire impostato da quello CAL premendo i tasti [SHIFT] [↓] oppure [SHIFT] [↑] . In questo modo, potete inserire i valori degli elementi di matrice per poi eseguire il calcolo matriciale, oltre a poter far eseguire al computer operazioni sulle matrici e fargli visualizzare gli elementi di matrice introdotti.

I tasti e le funzioni utilizzabili per introdurre e visualizzare gli elementi di matrice sono descritti di seguito.

Tasto	Funzione
[SHIFT] [↓]	• Imposta il modo MATRIX del computer • Vi permette di introdurre gli elementi della matrice X e quindi gli elementi della matrice Y e di far calcolare al computer le matrici. • Quando questi tasti sono premuti una seconda volta, il modo MATRIX del computer viene abbandonato.
[SHIFT] [↑]	• Imposta il modo MATRIX del computer • Permette al computer di effettuare operazioni sulle matrici. • Quando questi tasti sono premuti una seconda volta, il modo MATRIX del computer viene abbandonato.
[ENTER]	• Memorizza i numeri delle righe e delle colonneche formano una matrice oltre ad altri dati sugli elementi di matrice, dopodichè il computer attende la successiva inserzione dati.
[►]	• Sposta di una colonna verso destra il cursore (Quando il cursore si trova in corrispondenza dell'ultima colonna a destra, passa all'elemento successivo).
[◄]	• Sposta di una colonna verso sinistra il cursore (Quando il cursore si trova in corrispondenza dell'ultima colonna a sinistra, passa all'elemento precedente).
[↑]	• Sposta il cursore di una riga verso l'alto (cioè lo porta all'elemento immediatamente sopra la colonna attuale) • Riporta il computer alla fase di calcolo immediatamente precedente.
[↓]	• Sposta il cursore di una riga verso il basso (cioè lo porta all'elemento immediatamente sotto la colonna attuale) • Mette il computer in stato di attesa per la fase successiva dei calcoli.

Quando introducete i rispettivi elementi di una matrice, potete fare uso di qualsiasi tasto usato nel modo CAL per le quattro operazioni base ed il calcolo scientifico.

Esempio 1: Per inserire le seguenti due matrici

$$X = \begin{bmatrix} 10/3 & -5 & 2 \\ 8 & 2 & 23 \end{bmatrix}$$

$$Y = \begin{bmatrix} 5/3 & 3 & 2 \\ -1 & 0 & -8 \end{bmatrix}$$

Operazione:

[SHIFT] [↓]	MATRIX:X(Ø _, Ø)

Dato che la matrice X non è definita, sul visore compare (0,0) nel momento in cui il modo MATRIX viene impostato.

2	MATRIX:X(2 _, Ø)

"2" viene inserito come numero di righe.

[ENTER]	MATRIX:X(2, Ø _)
3 [ENTER]	X(1,1) Ø.

Inserite poi "3" come numero di colonne e definite poi la matrice X come una matrice di dimensione (2,3), dopodichè il computer è pronto a ricevere da voi il valore dell'elemento $X(1,1)$

10 [÷]	X(1,1) 1Ø.
3 [=]	X(1,1) 3.333333333
[ENTER]	X(1,2) Ø.

Una vo;ta che avete introdotto l'elemento $X(1,1)$, il computer attende che introduciate l'elemento successivo $X(1,2)$.

5 [+/−]	X(1,2) −5
[ENTER] 2	X(1,3) 2.
[ENTER] 8	X(2,1) 8.
[ENTER] 2	X(2,2) 2.
[ENTER] 23	X(2,3) 23.
[ENTER]	MATRIX:Y(Ø _, Ø)

Dopo che l'inserimento di tutti i dati sugli elementi della matrice X, dovete definire le dimensioni della matrice Y e quindi inserire i dati sugli elementi che ad essa appartengono nello stesso modo che per la matrice X.

2 [ENTER]	MATRIX:Y(2, Ø _)
3	MATRIX:Y(2, 3 _)
[ENTER] 5 [÷] 3 [=]	Y(1, 1) 1.666666667
[ENTER] 3	Y(1, 2) 3.
[ENTER] 2	Y(1, 3) 2.
[ENTER] 1 [+/−]	Y(2, 1) −1.
[ENTER] 0	Y(2, 2) Ø.
[ENTER] 8 [+/−]	Y(2, 3) −8.
[ENTER]	MATRIX OPERATION

Dopo l'inserimento di tutti gli elementi della matrice X sul visore appare il messaggio "MATRIX OPERATION", ad indicare che il computer è pronto ad eseguire calcoli con le matrici.

Se dovete calcolare la sola matrice X, premete [C·CE] [ENTER] nel momento in cui inserite il numero di righe e colonne, rispettivamente, della matrice Y.

Calcoli con le matrici

Premendo uno dei tasti elencati di seguito mentre sul visore appare il messaggio "MATRIX OPERATION", il computer esegue sulla matrice l'operazione che a tale tasto corrisponde.

Tasto	Funzione
[+]	$X + Y \rightarrow X$: Esegue l'addizione. Il risultato dell'addizione della matrice X a quella Y costituisce una nuova matrice X. Per poter eseguire un'addizione, le matrici X ed Y devono essere composte da un numero uguale sia di righe che di colonne.
[−]	$X - Y \rightarrow X$: Esegue la sottrazione. Il risultato della sottrazione di alcuni elementi della matrice Y da quella X costituisce gli elementi della nuova matrice X. Per poter eseguire una sottrazione, le matrici X ed Y devono essere composte da un numero uguale sia di righe che di colonne. (Esempio) $\begin{bmatrix} 2 & 3 \\ 5 & 2 \end{bmatrix} - \begin{bmatrix} 1 & 6 \\ 3 & -1 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & -3 \\ 2 & 3 \end{bmatrix}$
[×]	$X \cdot Y \rightarrow X$: Esegue la moltiplicazione. Per poter eseguire una moltiplicazione, il numero delle colonne della matrice X deve essere pari al numero di righe della matrice Y.
[÷]	$X \cdot Y^{-1} \rightarrow X$: Esegue la moltiplicazione della matrice X con la matrice Y inversa. Per poter eseguire tale moltiplicazione, il numero delle colonne della matrice X deve essere pari al numero di righe della matrice Y^{-1}.
[1/x]	$X^{-1} \rightarrow X$: Esegue il calcolo della matrice inversa X^{-1}. Il risultato dell'operazione diventa la nuova matrice X. Per poter eseguire questa operazione, la matrice X deve essere una matrice quadrata (cioè avente lo stesso numero di righe e colonne).
n [+]	$n + X \rightarrow X$: Esegue l'addizione dello scalare n agli elementi della matrice X. In questa operazione, lo scalare n viene aggiunto a ciascun elemento della matrice X. **NOTA:** Matematicamente, tale operazione non esiste. L'addizione di scalari è una una caratteristica esclusiva del computer.

Tasto	Funzione
n [−]	n − $X \rightarrow X$: Esegue la sottrazione degli elementi dello scalare n dagli elementi della matrice X. In questa operazione, allo scalare n viene sottratto ciascuno degli elementi della matrice X, ed il risultato diviene l'insieme degli elementi della nuova matrice X. **NOTA**: Matematicamente, tale operazione non esiste. La sottrazione di scalari è una una caratteristica esclusiva del computer. (Esempio) $2-\begin{bmatrix} 1 & 6 \\ 3 & -1 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & -4 \\ -1 & 3 \end{bmatrix}$
n [×]	n · $X \rightarrow X$: Esegue la moltiplicazione degli elementi della matrice X per lo scalare n.
n [÷]	n · $X^{-1} \rightarrow X$: Esegue la moltiplicazione degli elementi della matrice inversa X^{-1} per lo scalare n. Perchè questa operazione sia possibile, la matrice X^{-1} deve essere quadrata.
[↕]	$X \longleftrightarrow Y$: Scambia fra loro le due matrici X ed Y.
[T]	$X^t \rightarrow X$: Esegue la trasposizione della matrice X dando la nuova matrice trasposta come nuova matrice X.
[D]	$\lvert X \rvert \rightarrow X$ (Visore): Esibisce il valore del determinante della matrice X. Perchè questa operazione sia possibile, la matrice X deve essere quadrata.
[+/−]	$-X \rightarrow X$: Rovescia il segno positivo o negativo di ciascun elemento della matrice X.
[X^2]	$X \cdot X \rightarrow X$: Eleva al quadrato la matrice X. Perchè questa operazione sia possibile, la matrice X deve essere quadrata.
[X→M]	$X \rightarrow M$: Memorizza il valore della matrice X nella locazione di memoria della matrice M (cancellando il contenuto precedente della matrice M). Questo tasto viene usato per conservare il valore della matrice X anche dopo il termine dei calcoli.

Tasto	Funzione
[RM]	$M \rightarrow X$: Richiama il contenuto della matrice M nella matrice X (cancellando nel contempo il contenuto precedente della matrice X stessa).
[M+]	$X + M \rightarrow M$: Aggiunge il valore della matrice X cumulativamente al contenuto della memoria della matrice M. Perchè questa operazione sia possibile, le matrici X ed M devono essere uguali nel numero sia di righe che di colonne.

Note :
- Premendo il tasto [BRK] durante il calcolo di una matrice, tale calcolo viene sospeso. Ciò fatto i valori ritenuti delle matrici X, Y ed M saranno quelli precedenti l'inizio dell'operazione interrotta.
- Per eseguire la divisione degli elementi della matrice X per lo scalare n, premete i tasti [1/x] [X] in tale ordine.
- Se la maggior parte degli elementi della matrice hanno lo stesso valore, eseguite l'operazione n [+] con tutti gli eleenti della matrice fissati a zero e correggete poi solo il valore degli elementi di valore diverso da n. Ciò facilita l'inserimento dei dati sugli elementi della matrice.

Completato il calcolo della matrice, il messaggio "MATRIX CALCULATION" appare una seconda volta sul visore ad indicare che il computer è pronto per un altro calcolo di matrice. Dopo che il valore del determinante della matrice X è stato visualizzato premendo il tasto [D] , il messaggio "MATRIX OPERATION" riappare se viene premuto uno dei seguenti tasti : [↓] , [↑] , [◄] , [►] , [C·CE] , [ENTER] .

Esempio 2: Per calcolare $X + Y$, servendosi dei valori dei rispettivi elementi (presenti in memoria) delle matrici introdotte nell'Esempio 1.

$$X = \begin{bmatrix} 10/3 & -5 & 2 \\ 8 & 2 & 23 \end{bmatrix} \qquad Y = \begin{bmatrix} 5/3 & 3 & 2 \\ -1 & 0 & -8 \end{bmatrix}$$

Esecuzione :

	MATRIX OPERATION
[+]	**X+Y→X**
	(Il messaggio "BUSY" compare sul visore ad indicare che il computer sta eseguendo una operazione)
	MATRIX OPERATION

Vediamo ora qual'è il risultato dell'addizione :

[↓]	MATRIX:X(2 _ , 3)
[↓]	X(1, 1) 5.
[►]	X(1, 2) −2.
[SHIFT] [↓]	Ø.

Il modo MATRIX del computer viene ora abbandonato.

Se premete un qualsiasi tasto numerico oppure il tasto [•] mentre il messaggio "MATRIX OPERATION" è presente sul visore, il computer può eseguire operazioni scalari.

Esempio 3: Per calcolare 1/25* $X \rightarrow X$ servendosi del risultato del calcolo della matrice X nell'Esempio 2

Esecuzione :

[SHIFT] [↑]	MATRIX OPERATION
2	SCALAR 2.
5	SCALAR 25.
[1/X]	SCALAR Ø.Ø4
[×]	Ø.Ø4*X→X
	MATRIX OPERATION
[↓]	MATRIX:X(2 _ , 3)
[↓]	X(1, 1) Ø.2
[SHIFT] [↑]	Ø.

Esempio 4: Per risolvere le seguenti equazioni lineari simultanee a tre incognite servendosi del calcolo matriciale.

$$\begin{cases} 2x + 5y - z = -1 \\ x - y + 4z = 12 \\ 3x + 2y + z = 9 \end{cases}$$

SUGGERIMENTI: Inserite le matrici X ed Y indicate qui sotto e calcolate $X^{-1} \cdot Y$ per ottenere le soluzioni x,y e z delle equazioni.

$$X = \begin{bmatrix} 2 & 5 & -1 \\ 1 & -1 & 4 \\ 3 & 2 & 1 \end{bmatrix} \qquad Y = \begin{bmatrix} -1 \\ 12 \\ 9 \end{bmatrix}$$

Esecuzione :

Premete i tasti [SHIFT] e [↓] per impostare il modo MATRIX del computer e quindi introducete gli elementi di matrice di X ed Y dell'Esempio 1.

	MATRIX OPERATION	
[1/X]	**invX→X**	
	MATRIX OPERATION	
[X]	**X*Y→X**	
	MATRIX OPERATION	
[↓]	**MATRIX:X(3_, 1)**	
[↓]	**X(1, 1)**	**3.**
[↓]	**X(2, 1)**	**−1.**
[↓]	**X(3, 1)**	**2.**

Le soluzioni di x, y e z sono quindi le seguenti.

$x = 3,\ y = -1,\ z = 2$

Nota: Il calcolo di matrici si basa sul sistema, in largo uso, di eliminazione. Tuttavia, a causa della natura dei calcoli numerici di qualsiasi computer, potrebbero aversi errori nel calcolo di un determinante o di una matrice inversa a causa di troncamenti o per altre ragioni.

Esempio 5: Risolvere la matrice inversa di $\begin{bmatrix} 3 & 1 \\ 1 & 1/3 \end{bmatrix}$

Questa non è una matrice regolare e non possiede quindi teoreticamente una matrice inversa. In qualsiasi computer, tuttavia, il valore 1/3 viene inserito come 0,33 ...3 ed una matrice inversa quindi esiste, e da quindi i seguenti risultati.

$$\begin{bmatrix} 3 & 1 \\ 1 & 0.33\cdots3 \end{bmatrix}^{-1} = \begin{bmatrix} -33\cdots3. & 1.E10 \\ 1.E10 & -3.E10 \end{bmatrix}$$

I risultati ottenuti da un calcolatore possono perciò, in tale modo, essere errati. Notate che la verifica con un qualsiasi altro metodo può essere necessaria a seconda di come il calcolo di matrici viene applicato. Nell'esempio qui sopra, quando ottenete il valore del determinante moltiplicando la matrice originale X per 3, potete scoprire che la matrice X non è regolare poichè il risultato della moltiplicazione risulta essere pari a zero($\begin{vmatrix} 9 & 3 \\ 3 & 1 \end{vmatrix} = 0$).

Nota: Dato che il calcolo di una matrice non può venire completato con una sola operazione (ad es. una sola moltiplicazione) il completamento dei calcoli richiede un certo tempo. La soluzione di una matrice inversa di una matrice consistente in 7 righe e 7 colonne richiede circa 6 secondi. Questo tempo di calcolo varia a seconda dei valori delle matrici.

Capacità di memoria richiesta dai calcoli di matrice

- Dato che i calcoli di matrici occupano la stessa area di memoria dei programmi BASIC, la capacità di memoria non utilizzata (cioè la capacità determinabile da MEM [ENTER] nel modo BASIC) deve rimanere superiore a quella espressa dal risultato della seguente formula.

[(No. di righe della matrice X) × (No. di colonne della matrice X) × 8 + 7] byte
+ [(No. di righe della matrice Y) × (No. di colonne della matrice Y) × 8 + 7] byte
+ [(No. di righe della matrice M) × (No. di colonne della matrice M) × 8 + 7] byte
+ [(No. di righe della matrice risultante) × (No. delle colonne della matrice risultante) × 8 + 7] byte

Tuttavia, quando nè la matrice Y nè quella M vengono usate, i valori fra parentesi (cioè il numero di righe e colonne) di ciascuna delle matrici inutilizzate viene considerato pari a zero nel calcolo della capacità di memoria. La matrice risultante è necessaria solo durante il calcolo e viene cancellata quando questo termina. Fra l'altro, nessuna matrice risultante è richiesta per l'esecuzione di [X↔M], [RM] o [↕]. Due matrici risultanti sono necessarie per operazioni su matrici impieganti i tasti [÷] ed n [÷], dato che queste operazioni comportano due calcoli (cioè inversione e moltiplicazione).

- Se mentre vi trovate nel modo MATRIX sul display appare il messaggio "MEMORY OVER", cancellate le variabili ed i programmi del mado BASIC per aumentare l'area di memoria non utilizzata impiegabile per i calcoli di matrice.
  Se eseguite calcoli su matrici dotate di un gran numero di elementi, usate una scheda RAM di dimensioni maggiori.

Esempio 6: Eseguire la moltiplicazione dele seguenti due matrici ($X \cdot Y \rightarrow X$)

$$X = \begin{bmatrix} 2 & 3 \\ 5 & 1 \end{bmatrix} \quad Y = \begin{bmatrix} 8 & 30 \\ 7 & 15 \end{bmatrix}$$

(la matrice M non è definita)

La capacità di memoria necessaria si calcola nel modo seguente :

$$\underbrace{(2 \times 2 \times 8 + 7)}_{\text{Matrice } X} + \underbrace{(2 \times 2 \times 8 + 7)}_{\text{Matrice } Y} + \underbrace{(2 \times 2 \times 8 + 7)}_{\text{Matrice risultante}} = 117 \text{ byte}$$

Stampa di matrici

Per stampare i dati (cioè il valore di ciascuno degli elementi) della matrice X, preparate ed eseguite il seguente programma. Se eseguite il programma inserendo"RUN"e premendo poi il tasto [ENTER] , tuttavia, ogni dato relativo alle matrici verrà cancellato dalla memoria. Siate certi quindi di eseguire il programma usando solo il tasto [DEF] .

```
100 "M":INPUT "ROW";II
110 INPUT "COLUMN";JJ
120 FOR I=0 TO II-1
130 FOR J=0 TO JJ-1
140 LPRINT "X(";I+1;",";J+1;")=;"X(I,J)
150 NEXT J:NEXT I:END
```

Se volete stampare i dati delle matrici Y o M, cambiate i due simboli "X" della riga 140 del programma in altrettanti "Y" o "M".

Esecuzione: Premete i tasti [DEF] [M] nel modo RUN e la matrice designata verrà mandata alla stampante.

Messaggi di errore

Se durante il calcolo di una matrice viene commesso un errore, sul visore compaiono i seguenti messaggi di errore accompagnati dall'indicazione "E" (errore). Per abbandonare la condizione di errore, premete il tasto [C-CE] e l'indicazione "E" scompare lasciando il posto al messaggio "MATRIX OPERATION". I dati sul calcolo di matrice in fase di soluzione nel momento in cui si è commesso un errore sono ancora presenti in memoria.

Messaggio di errore	Causa dell'errore
IMPOSSIBLE CALCULATION (CALCOLO IMPOSSIBILE)	• Le matrici non sono di dimensione uguale. Le matrici addizionate, sottratte o moltiplicate non sono di dimensione uguale, oppure si è tentato il calcolo della matrice inversa o del quadrato di una matrice non quadrata.
MEMORY OVER (MEMORIA ESAURITA)	• Memoria insufficiente Nell'operazione $X \rightarrow M$, lo spazio di memoria si è rivelato insufficiente per memorizzare la matrice M, oppure non esistono aree a disposizione per l'operazione aritmetica.
DIVISION BY ZERO (DIVISIONE PER ZERO)	• Lo 0 (zero) viene usato come divisore. Si è tentato di usare lo zero per la divisione di un numero nel calcolare una matrice inversa.
OVER FLOW (ECCEDENZA DI CAPACITÀ)	• Nel corso di una operazione aritmetica si è avuto un'eccedenza di capacità.

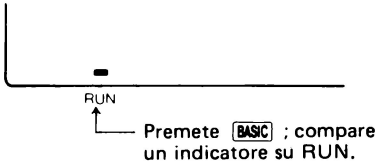
Calcolo manuale in BASIC

Cosa è il calcolo manuale

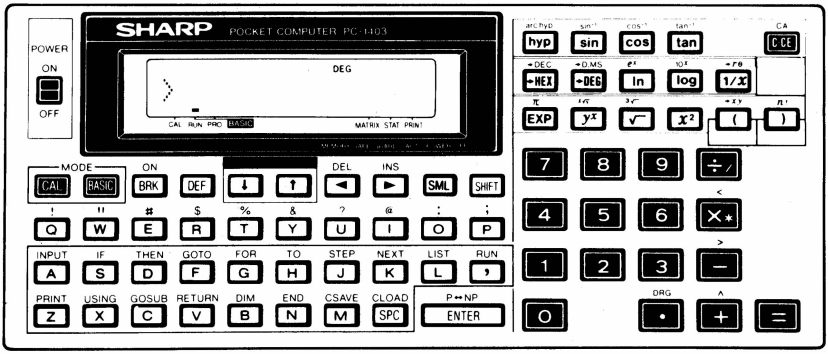
Il computer può essere usato fondamentalmente in due modi. In un modo potete memorizzare anticipatamente l'intera procedura di calcolo o i diversi passaggi nella memoria del computer, e quindi lasciare che il computer esegua i calcoli automaticamente. Nell'altro modo, potete eseguire i calcoli passaggio per passaggio usando manualmente i tasti, controllando nel frattempo i risultati intermedi dei calcoli. Quest'ultimo modo viene chiamato il "calcolo manuale".
Naturalmente nel modo CAL, tutti i calcoli vengono eseguiti manualmente. Qui invece chiamiamo calcolo manuale soltanto il calcolo manuale eseguito nel modo BASIC (modo RUN o PROGRAM).

Come eseguire manualmente i calcoli

Proviamo ora un calcolo manuale nel modo RUN. Premete il tasto BASIC per mettere il computer nel modo RUN.



Nel modo RUN, sono operativi i tasti e le funzioni mostrati nella figura seguente (altrettanto vale per il modo PROGRAM):



Prima di passare agli esempi, vediamo alcuni dei punti importanti da ricordare.

Mentre generalmente usiamo gli operatori +, –, x, ÷: per i calcoli matematici su carta, non usiamo gli operatori x e ÷ per le operazioni aritmetiche nel BASIC. Invece del x e del ÷ usiamo rispettivamente un asterisco (*) ed una barra (/).

Gli operatori * e / possono essere immessi premendo rispettivamente i tasti [X*] e [÷/].

Per ottenere il risultato del calcolo manuale, usate il tasto [ENTER] invece del tasto [=]. Non usate il segno di dollari o le virgole quando scrivete i calcoli con il **computer**. Questi caratteri hanno un significato speciale nel linguaggio di programmazione BASIC.

Vediamo ora dei semplici esempi aritmetici. Ricordate di azzerare la macchina tra i calcoli con il tasto [C·CE].

Premere	Visore
[5] [Ø] [+] [5] [Ø] [ENTER]	1ØØ.
[1] [Ø] [Ø] [–] [5] [Ø] [ENTER]	5Ø.
[6] [Ø] [*] [1] [Ø] [ENTER]	6ØØ.
[3] [Ø] [Ø] [/] [5] [ENTER]	6Ø.
[1] [Ø] [SHIFT] [^] [2] [ENTER]	1ØØ.
[2] [*] [SHIFT] [π] [ENTER]	6.283185307
[√] [6] [4] [ENTER]	8.

Richiamo dei dati immessi

Anche dopo che il **computer** ha visualizzato i risultati dei vostri calcoli, potete richiamare sul visore l'ultimo dato da voi immesso. Per far ciò, usate le frecce con l'indicazione a sinistra [◄] e con l'indicazione a destra [►].

La freccia a sinistra [◄] richiama l'espressione con il cursore posizionato dopo **l'ultimo** carattere.

La freccia a destra [►] richiama l'espressione con il cursore posizionato "sopra" al **primo** carattere.

Ricordate che le frecce a sinistra ed a destra vengono usate anche per posizionare il cursore lungo una riga. Le frecce a destra ed a sinistra sono molto utili per rivedere (o modificare) i dati immessi senza dover ribattere l'intera espressione.

Imparerete ora ad usare le frecce a destra ed a sinistra negli esempi seguenti. Immaginate di essere il responsabile marketing di una società ed eseguite i calcoli di mano in mano che si presentano.

Nella vostra qualità di responsabile del marketing, dovete pianificare l'incontro annuale con la forza di vendita. Prevedete che 300 persone prenderanno parte a questo incontro della durata di 3 giorni. Per una parte di questo periodo, la forza di vendita seguirà i lavori in piccoli gruppi. Ritenete che sarebbero appropriati dei gruppi di 6 persone. Quanti gruppi si otterrebbero?

Premete	Visore
[3] [0] [0] [/] [6] [ENTER]	50.

Ripensandoci, decidete che sarebbe meglio formare dei gruppi contenenti un numero dispari di partecipanti. Richiamate l'ultimo dato da voi immesso usando la freccia [◄].

Premete	Visore
[◄]	300/6_

Per calcolare il nuovo numero di gruppi, dovete sostituire il 6 con un numero dispari. 5 sembra meglio di 7. Dal momento che avete richiamato il dato usando la freccia [◄], il cursore è posizionato all'estremità del visore. Usate la [◄] per spostare il cursore di uno spazio a sinistra.

Premete	Visore
[◄]	300/6

Notate che dopo che avete spostato il cursore, questo diventa un blocco lampeggiante . Ogni qualvolta posizionate il cursore "sopra" ad un carattere esistente, questo viene visualizzato con il cursore lampeggiante.

Premete il 5 per sostituire il 6. Attenzione, quando battete un nuovo carattere su uno esistente, quello originale viene perso! Non potete richiamare un'espressione che avete modificato.

Premete	Visore
[5]	300/5_
[ENTER]	60.

Sessanta sembra un numero ragionevole di gruppi, pertanto decidete che ciascun gruppo sarà costituito da 5 partecipanti.

La funzione di richiamo è utile anche per verificare l'ultimo dato da voi immesso, particolarmente quando i risultati ottenuti non sembrano logici. Per esempio, supponiamo che abbiate eseguito questo calcolo:

Premete	Visore
[3] [0] [/] [5] [ENTER]	6.

Persino un manager stanco e stressato come voi si rende conto che 6 non é un risultato ragionevole dal momento che si tratta di centinaia di persone! Richiamate l'ultimo dato immesso usando la [►] .

Premete	Visore
[►]	30/5

Poiché avete richiamato il dato usando la [►] , il cursore lampeggiante è ora posizionato sul primo carattere del visore. Per correggere questo dato, desiderate aggiungere uno zero. Usando la [►] , spostate il cursore sino a che questo è posizionato sullo zero. Quando effettuate un INSerimento, posizionate il cursore lampeggiante sul carattere che seguirà quello in cui desiderate effettuare l'inserimento.

Premete	Visore
[►]	30/5

Usate il tasto di INSerimento per far spazio per la cifra mancante.

Premete	Visore
[SHIFT] [INS]	3_0/5

Premendo INSerimento si spostano tutti i caratteri di uno spazio a destra, e si inserisce uno spazio libero tra parentesi. Il cursore lampeggiante è ora posizionato su questo spazio libero, ad indicare la posizione del nuovo dato. Premete lo zero. Una volta corretto il dato immesso, visualizzate il vostro nuovo risultato.

Premete	Visore
[0]	300/5
[ENTER]	60.

D'altra parte, supponiamo che abbiate impostato questo calcolo:

Premete	Visore
[3] [0] [0] [0] [/] [5] [ENTER]	600.

Il risultato sembra eccessivamente alto. Le persone che partecipano al convegno sono soltanto 300, come è possibile che vi siano 600 "piccoli gruppi"? Richiamate il dato immesso usando la [►].

Premete	Visore
[►]	3000/5

Il cursore lampeggiante è ora posizionato sul primo carattere del visore. Per correggere questo dato, eliminate uno degli zeri. Usando la [►] spostate il cursore sul primo zero (o su qualsiasi zero). Quando cancellate un carattere, dovete posizione il cursore "sopra" il carattere da cancellare.

Premete	Visore
[►]	3000/5

Usate ora il tasto DEL (cancellazione) per eliminare uno degli zeri.

Premete	Visore
[SHIFT] [DEL]	300/5

Premendo il tasto DEL, si provoca lo spostamento a sinistra di uno spazio di tutti i caratteri. In questo modo si cancella il carattere "sopra" al quale si trova il cursore nonché lo spazio occupato dal carattere stesso. Il cursore lampeggiante rimane nella stessa posizione ad indicare la posizione successiva per l'immissione. Poiché non dovete effettuare altri cambiamenti, completate il calcolo.

Premete	Visore
[ENTER]	6Ø.

(Nota: Premendo il tasto SPC (spazio), quando il cursore è posizionato su di un carattere, si sostituisce il carattere con uno spazio in bianco. Il tasto DEL (cancellazione) elimina viceversa il carattere e lo spazio da esso occupato).

Errori

La possibilità di richiamare l'ultimo dato immesso è essenziale quando compare il famigerato messaggio ERROR. Immaginiamo che, non volutamente, abbiate battuto questo dato:

Premete	Visore
[3] [Ø] [Ø] [/] [/] [5] [ENTER]	ERROR 1

Naturalmente siete sorpresi quando compare questo messaggio! ERROR 1 è semplicemente il modo in cui il computer vi segnala di non capire cosa si voglia ottenere da lui. Per scoprire qual è il problema, richiamate l'ultimo dato immesso usando le frecce [◄] o [►].

Premete	Visore
[◄]	3ØØ//5

Sia che usiate il tasto [◄] o [►], il cursore lampeggiante indica il punto in cui il computer non capisce cosa fare. Non c'è da meravigliarsi, avete immesso troppi operatori! Per correggere questo errore usate il tasto DEL (cancellare).

Premete	Visore
[SHIFT] [DEL] [ENTER]	6Ø.

Se, dopo aver richiamato il dato dopo un messaggio ERROR 1, scoprite che avete **omesso** un carattere, usate la sequenza di INSerimento per correggerlo.

Quando usate il **computer** come calcolatrice, la maggioranza degli errori che incontrate sarà un ERROR 1 (un errore di sintassi). Per l'elenco completo dei messaggi di errore, vedete l' APPENDICE A.

Calcoli in serie

Il **computer** vi permette di riprendere i risultati di un calcolo per successive operazioni come vedremo qui di seguito.

Ritorniamo all'esempio della conferenza e supponiamo che uno dei Vostri compiti sia quello di preparare un dettagliato budget delle spese per questo metting. Sapete che avete a disposizione 15Ø $ per ogni partecipante, pertanto il Vostro budget totale sarà:

Premere	Visore
[3] [Ø] [Ø] [*] [1] [5] [Ø] [ENTER]	45ØØØ.

Di questa cifra pensate che il 15% sarà utilizzato per la serata finale da trascorrersi in un buon locale per la premiazione del miglior venditore. Pertanto il risultato della precedente operazione Vi serve per effettuare questo calcolo. In questi casi NON PREMETE il tasto [C·CE] ma operate come segue:

Premere	Visore
[*] [•] [1] [5]	45ØØØ.*.15_

Come potete notare il risultato della precedente operazione è diventato un elemento del nuovo calcolo. In tutti i casi nei quali effettuate dei calcoli in serie, la seconda operazione e le successive devono cominciare con un operatore aritmetico (+ – /*) o, con il simbolo d'elevamento a potenza. Come al solito per comandare l'esecuzione premete [ENTER].

Attenzione: I tasti [%] e [Δ%] non sono utilizzabili per i calcoli percentuali. Questo tasto ha solo funzioni grafiche. Il tasto [%] deve essere usato solo come carattere e il tasto [Δ%D] rimane inoperativo.

Esempio: 45ØØØ [*] 15 [SHIFT] [%] → ERROR 1

Premere	Visore
[ENTER]	675Ø.

Continuando nell'esame del Vostro budget sapete che da questa somma dovete togliere 4000 $ per il rinfresco.

Premere	Visore
[−] [4] [0] [0] [0]	6750.−4000_
[ENTER]	2750.

Le decorazioni costeranno 1225 $.

Premere	Visore
[−] [1] [2] [2] [5] [ENTER]	1525.

Inoltre il presentatore e l'orchestra Vi costeranno 2200 $.

Premere	Visore
[−] [2] [2] [0] [0] [ENTER]	−675.

Non Vi resta ora che modificare le Vostre spese o trovare ulteriori fonti di finanziamento.

Numeri negativi

Poiché volete che la serata conclusiva sia veramente speciale, decidete di spendere questa somma in più. Ora volete anche sapere quale è la nuova percentuale rispetto alla somma totale a disposizione con questa somma aggiuntiva.
Prima di tutto bisogna cambiare il segno:

Premere	Visore
[*] [−] [1]	−675.*−1_
[ENTER]	675.

Ora aggiungete questo importo al Vostro precedente preventivo di spesa.

Premere	Visore
[+] [6] [7] [5] [0] [ENTER]	7425.

Dividendo per 45000 otterrete la percentuale di questo importo sul Vostro budget.

Premere	Visore
[/] [4] [5] [0] [0] [0] [ENTER]	0.165

Allora decidete di roiservare il 16,5% alla cerimonia di premiazione.

Calcoli composti e parentesi

Molti dei calcoli sopra descritti si sarebbero potuti combinare su di una sola riga come si può notare dall'esempio che segue:

675+6750/45000

Quando s'impostano questi calcoli bisogna stare molto attenti; infatti:

675+6750/45000 può essere interpretato:

$$\frac{675+6750}{45000} \quad \text{o} \quad 675+\frac{6750}{45000}$$

Il **computer** svolge operazioni composte seguendo particolari regole di valutazione delle espressioni secondo determinati ordini di priorità (vedere pagina 77). AssicurateVi d'impostare i calcoli che Voi volete vengano effettuati utilizzando le parentesi per meglio chiarire le Vostre espressioni.

(675+6750) / 45000 oppure 675+(6750 / 45000)

Per meglio comprendere l'uso delle partentesi svolgete i seguenti esempi:

Premete	Visore
[(] [6] [7] [5] [+] [6] [7] [5] [0] [)] [/] [4] [5] [0] [0] [0] [ENTER]	0.165
[6] [7] [5] [+] [(] [6] [7] [5] [0] [/] [4] [5] [0] [0] [0] [)] [ENTER]	675. 15

Nota: Nel modo BASIC (PRO or RUN) la chiusura di parentesi non può essere omessa prima del tasto [ENTER] . Nel modo CAL, tuttavia, tale omissione è possibile prima del tasto [=] .

L'uso delle variabili nei calcoli

Il computer può memorizzare 26 variabili fisse con le lettere da A a Z. Se non avete dimestichezza con il concetto di variabile, leggete le spiegazioni del capitolo 4. Le variabili vengono designate per mezzo dell'istruzione di assegnazione.

A = 5
B = –2

Oltre ad assegnare direttamente un valore, possiamo anche assegnare il valore di una variabile ad un altra variabile:

C = A + 3
D = E

Pertanto la nostra variabile può sostituire un numero in un calcolo come meglio vediamo qui di seguito.

Ora che avete programmato l'ultima serata desiderate completare il budget di tutta la conferenza. Inoltre volete calcolare anche tutte le percentuali di spesa.

Prima di tutto dovete trovare qual'è la somma ancora disponibile e, questo valore assegnarlo alla variabile R.

Premere	Visore
[R] [=] [4] [5] [Ø] [Ø] [Ø] [–] [7] [4] [2] [5]	R=45ØØØ–7425_
[ENTER]	37575.

Non appena premete il tasto [ENTER], il computer esegue il calcolo ed assegna il risultato alla variabile R e, per visualizzare il valore di una variabile (calcolato o assegnato in precedenza), basta scrivere il nome della variabile e premere il tasto [ENTER] come nel seguente esempio:

Premere	Visore
[R] [ENTER]	37575.

Ora effettuiamo dei calcoli utilizzando le variabili. Attenzione, il valore di R non cambia fino a che Voi non le assegnerete un nuovo valore.
Ora tornando al nostro esempio, desiderate destinare il 60% della somma rimanente al noleggio della sala:

Premere	Visore
[R] [*] [•] [6] [Ø]	R*.6Ø_
[ENTER]	22545.

Ed il 25% alle spese di direzione e training del seminario.

Premere	Visore
[R] [*] [•] [2] [5] [ENTER]	9393.75

Le variabili ricordano il loro contenuto anche a macchina spenta e nel caso intervenga l'autospegnimento.
Il contenuto delle variabili si perde solo nei seguenti casi:

* Viene assegnato un nuovo valore alla variabile.
* Scrivete la parola CLEAR e premete [ENTER] (non il tasto [C·CE] che cancella solo il visore).
* Viene azzerata la macchina premendo il pulsante ALL RESET.
* Vengono cambiate le pile.

Vi sono certe limitazioni nell'assegnazione delle variabili e certe procedure di programmazione che ne provocano il cambiamento. Consultate il capitolo 4 per quanto concerne l'assegnazione ed il capitolo 5 per l'uso delle variabili nella programmazione.

Calcoli in catena

Oltre a combinare parecchi operatori in un unico calcolo, il **computer** vi consente di eseguire parecchi calcoli uno di seguito all'altro senza dover premere il tasto [ENTER] prima di proseguire. Dovete separare le equazioni con delle virgole. Verrà visualizzato soltanto il risultato del calcolo **finale.** (Ricordatevi inoltre che la lunghezza massima di riga accettata dal computer è di 8Ø caratteri incluso il tasto [ENTER]).

Tornando al nostro esempio precedente, supponiamo che vogliate calcolare di quanto denaro avreste potuto disporre per le camere se aveste mantenuto l'assegnazione originale del 15% per la serata finale.

Premete	Visore
[R] [=] [.] [8] [5] [*] [4] [5] [Ø] [Ø] [Ø] [,] [R] [*] [.] [6] [Ø]	R=.85*45ØØØ,R*.6Ø_

Il computer esegue tutti i calcoli in catena ma visualizza soltanto il risultato finale.

Premete	Visore
[ENTER]	22950.

Per trovare il valore di R usato in questo calcolo, premete R:

Premete	Visore
[R] [ENTER]	38250.

Messaggio di errore

Se si è verificato un errore in conseguenza di un calcolo manuale, sul visore compare un messaggio di errore quale:

ERROR 1 o ERROR 2

Lo stato di errore può essere annullato con il tasto [C·CE] o [◄] o [►]. Se si usano i tasti [◄] o [►] per annullare lo stato di errore, la parte della formula in cui si è verificato l'errore viene richiamata sul visore (vedere la descrizione della funzione di richiamo).

Notazione scientifica

Coloro che lavorano con numeri molto grandi e molto piccoli, usano spesso un formato speciale chiamato notazione esponenziale o scientifica. Nella notazione scientifica, il numero è suddiviso in due parti.

La prima parte è costituita da un numero decimale regolare compreso tra 1 e 1Ø. La seconda parte rappresenta quanto il numero sia grande o piccolo in potenze di 1Ø.

Come sapete, il primo numero alla sinistra del punto decimale in un numero decimale regolare indica il numero delle unità (1), il secondo indica il numero delle decine (1Ø), il terzo il numero delle centinaia (1ØØ), ed il quarto il numero delle migliaia (1ØØØ). Sono semplicemente delle potenze crescenti di 1Ø:

$10^{\phi} = 1$, $10^{1} = 10$, $10^{2} = 100$, $10^{3} = 1000$, etc.

La notazione scientifica scompone un numero decimale in due parti: una indica quali sono le cifre, la seconda indica la distanza di un numero dalla sinistra o dalla sinistra o dalla destra del punto decimale. Per esempio:

1234 diventa 1.234 per 10^{3} (3 posizioni a destra)
654321 diventa 6.54321 per 10^{5} (5 posizioni a destra)
.000125 diventa 1.25 per 10^{-4} (4 posizioni a sinistra)

La notazione scientifica è utile per molte abbreviazioni. Pensate a cosa si dovrebbe scrivere per indicare $1{,}0 \times 10^{87}$ – un 1 ed 87 zeri! Viceversa nella notazione scientifica questo numero appare come segue:

1.0×10^{87} oppure **1.0E 87**

Il **computer** utilizza la notazione scientifica ogniqualvolta i numeri diventano troppo grandi da visualizzare usando la notazione decimale. Questo computer usa uno speciale simbolo di potenza, la **E** per indicare "per dieci alla":

1234567890000 viene visualizzato come 1.23456789 **E** 12
.000000000001 viene visualizzato come 1. **E** –12

Coloro che non sono molto pratici di questo tipo di notazione, devono dedicare qualche minuto per scrivere alcuni numeri molto grandi ed alcuni numeri molto piccoli per verificare come vengono visualizzati.

Limiti

Il numero più grande che il **computer** può trattare è constituito da 10 cifre significative, con esponenti di 2 cifre. In altre parole il numero più grande è:

9.999999999**E** 99 = 999999999900000000000000000000000000
000000000000000000000000000000000000
000000000000000000000000000000000

ed il numero più piccolo è:

9.999999999**E** –99 = .000000000000000000000000000000000000
000000000000000000000000000000000000
000000000000000000000000000000000009
999999999

In alcune circostanze, quando i numeri vengono usati frequentemente, il **computer** usa una speciale forma compatta. In questi casi vi sono dei limiti particolari imposti sulla dimensione dei numeri, generalmente da Ø a ~ 65535 o −32768 ~ 32767. Coloro che hanno una certa conoscenza dei computer riconosceranno questi due numeri come la gamma più vasta che può essere rappresentata in 16 bit binari. I casi in cui viene usata questa forma sono indicati nel capitolo 9.

Funzione ultimo risultato

Nel caso del calcolo in serie, potete usare il risultato del calcolo soltanto come primo membro della successiva formula di calcolo.
Guardate l'esempio seguente.

Premere	Visore
3 [+] 4 [ENTER]	7.
[*] 5	7.*5_
[ENTER]	35.

Premete [C·CE], quindi il tasto [↓] o [↑]. Se avete premuto questi tasti dopo aver terminato l'esempio di calcolo precedente, sul vostro visore è visualizzato "35.". Il dato numerico visualizzato non è nient'altro che il risultato del vostro ultimo calcolo.

Il **computer** è in grado di "ricordare" l'ultimo risultato ottenuto con il calcolo manuale, e di richiamarlo sul visore con il tasto [↓] o [↑].

Nel caso del calcolo in serie precedentemente descritto, potete usare il risultato del calcolo precedente soltanto come primo membro della formula di calcolo successiva. Invece, con la funzione: ultimo risultato, potete mettere il risultato del calcolo precedente in qualsiasi posizione del calcolo successivo.

(Esempio) Usate il risultato (6.25) dell'operazione: 50 ÷ 8, per calcolare
12 x 5 ÷ 6.25 + 24 x 3 ÷ 6.25 = :

Premere	Visore
50 [/] 8 [ENTER]	6.25
	Ultimo risultato
12 [*] 5 [/] [↑]	12*5/6.25_
	Ultimo risultato richiamato

Premere	Visore
[+] 24 [*] 3 [/] [↓]	12*5/6.25+24*3/6.25_
	Ultimo risultato richiamato
[ENTER]	21.12
[C·CE] [↓]	21.12_
	L'ultimo risultato viene sostituito dal risultato del calcolo precedente eseguendo un calcolo manuale con il tasto [ENTER] .

Come mostrato in questo esempio, l'ultimo risultato può essere richiamato in qualsiasi momento e in qualsiasi luogo, ma verrà sostituito con il nuovo ultimo risultato ottenuto dall'ultimo calcolo.

L'ultimo risultato non viene cancellato dall'uso dei tasti [C·CE] o [SHIFT] [CA].

Nota: L'ultimo risultato non può venire richiamato quando l'esecuzione del programma viene interrotta temporaneamente in un modo che non sia quello RUN o quando il programma viene eseguito nel modo TRACE.

Lunghezza delle formule

La lunghezza delle formule che potete scrivere con il vostro computer ha certi limiti. Con il **computer**, è possibile premere un massimo di 79 tasti per scrivere una singola formula di calcolo (escludendo il tasto [ENTER]). Se tentate di premere l'ottantesimo tasto, il cursore ( ) comincerà a lampeggiare su tale carattere, ad indicare che l'ottantesima immissione non è valida.

Calcoli scientifici nel modo BASIC

Questo computer ha molte funzioni scientifiche che possono essere usate nel modo BASIC.

Per eseguire le funzioni scientifiche dovete premere [ENTER] alla fine della scrittura, viceversa i vostri calcoli non verranno elaborati dal computer.

Queste funzioni verranno descritte come segue:

Funzioni	Scrittura	Operazione	Note
Funzioni trigonometriche			
sin	SIN	[sin]	
cos	COS	[cos]	
tan	TAN	[tan]	
Funzioni trigonometriche inverse			
$\sin^{-1}$	ASN	[SHIFT] [$\sin^{-1}$]	
$\cos^{-1}$	ACS	[SHIFT] [$\cos^{-1}$]	
$\tan^{-1}$	ATN	[SHIFT] [$\tan^{-1}$]	
Funzioni iperboliche			
sinh	HSN	[hyp] [sin]	
cosh	HCS	[hyp] [cos]	
tanh	HTN	[hyp] [tan]	
Funzioni iperboliche inverse			
$\sinh^{-1}$	AHS	[SHIFT] [arc hyp] [$\sin^{-1}$]	
$\cosh^{-1}$	AHC	[SHIFT] [arc hyp] [$\cos^{-1}$]	
$\tanh^{-1}$	AHT	[SHIFT] [arc hyp] [$\tan^{-1}$]	
Funzioni logaritmiche			
ln	LN	[ln]	$\log_e x$
log	LOG	[log]	$\log_{10} x$
Funzioni esponenziali			
e^x	EXP	[SHIFT] [e^x]	e ≒ 2.718281828
10^x	TEN	[SHIFT] [10^x]	
Reciproco $\frac{1}{x}$	RCP	[$1/x$]	
Quadrato x^2	SQU	[x^2]	
Radice quadrata $\sqrt{\ }$	$\sqrt{\ }$ o SQR	[$\sqrt{\ }$]	
Radice cubica $\sqrt[3]{\ }$	CUR	[SHIFT] [$\sqrt[3]{\ }$]	

Funzioni	Scrittura	Operazione	Note
Fattoriale n!	FACT	[SHIFT] [n!]	
Pi greco	π o PI	[SHIFT] [π]	$\pi \fallingdotseq$ 3.141592654
DMS → DEG	DEG	[→DEG]	
DEG → DMS	DMS	[SHIFT] [→DMS]	
Potenza y^x	^	[SHIFT] [^] o [y^x]	y ^ x : y^x
Radice xesima di y $\sqrt[x]{y}$	ROT	[SHIFT] [$^x\sqrt{y}$]	y ROT x : $\sqrt[x]{y}$
Coordinate rettangolari → Coordinate polari	POL	[SHIFT] [→rθ]	
Coordinate polari → Coordinate rettangolari	REC	[SHIFT] [→xy]	
Numero intero	INT	[I] [N] [T]	INT (x)
Assoluto $\lvert x \rvert$	ABS	[A] [B] [S]	ABS (x)
Segno	SGN	[S] [G] [N]	SGN (x) $x > 0$: 1 $x = 0$: 0 $x < 0$: −1
Modifica (arrotondamento)	MDF	[M] [D] [F]	

Di queste, le funzioni INT, ABS e SGN possono essere immesse usando i tasti alfabetici. Anche alcune altre funzioni possono essere immesse con i tasti alfabetici. Per esempio, "sin 30" può essere immesso sia premendo [sin] 30 che [S] [I] [N] 30. Per le funzioni trigonometriche e trigonometriche inverse, nonché per la conversione delle coordinate, è necessario specificare anticipatamente l'unità angolare voluta. Nel calcolo manuale, le unità angolari possono essere specificate sia premendo [SHIFT] [DRG] che con le seguenti istruzioni:

Unità angolare	Comando	Simbolo sul visore	Descrizione
Grado sessagesimale	DEGREE	DEG	Rappresenta un angolo retto in 90 [∘]
Radiante	RADIAN	RAD	Rappresenta un angolo retto in $\pi/2$. [rad]
Grado centesimale	GRAD	GRAD	Rappresenta un angolo retto in 100 [g]

Queste istruzioni vengono usate per specificare le unità angolari nel programma. Esercitatevi ad usare queste istruzioni per specificare le unità angolari nei seguenti esempi di calcolo:

(Esempio) sin 30° =

(Operazione) DEGREE [ENTER] (Specifica "gradi sessagesimali" come unità angolare)

SIN 30 [ENTER]

DEG 0.5

(Esempio) $\tan \frac{\pi}{4}$ =

(Operazione) RADIAN [ENTER] (Specifica "radianti" come unità angolare)

TAN (PI/4) [ENTER]

RAD 1.

(Esempio) $\cos^{-1}$ (–0.5) =

(Operazione) DEGREE [ENTER] (Specifica "grado sessagesimale" come unità angolare)

ACS – 0.5 [ENTER]

DEG 120.

(120°)

(Esempio) log 5 + ln 5 =

(Operazione) LOG 5 + LN 5 [ENTER]

2.308407917

(Esempio) e^{2+3} =

(Operazione) EXP (2 + 3) [ENTER]
(Non usate il tasto [EXP])

148.4131591

(Esempio) $\sqrt[3]{4^3 + 5^3} =$

(Operazione) CUR (4 ^ 3 + 5 ^ 3) [ENTER] **5.738793548**

(Esempio) Converte 30 gradi 30 minuti dalla notazione sessagesimale alla notazione decimale.

(Operazione) DEG 30. 30 [ENTER] **30.5**

(30.5 gradi)

(Esempio) Converte 30.755 gradi dalla notazione decimale in notazione sessagesimale.

(Operazione) DMS 30.755 [ENTER] **30.4518**

(30 gradi 45 min. 18 sec.)

(Esempio) Conversione dalle coordinate ortogonali alle coordinate polari: determinate le coordinate polari (r, θ) per il punto P (3, 8) in coordinate ortogonali:

(Operazione) DEGREE [ENTER] (Specifica "gradi" sessagesimali come unità angolare)

POL (3, 8) [ENTER] (r) **8.544003745**

(r ≒ 8.5)

Z [ENTER] (θ) **69.44395478**

(θ ≒ 69°)

* **Il valore di θ viene trasferito alla variabile Z, ed il valore di r alla variabile Y.**

(Esempio) Conversione dalle coordinate polari alle coordinate rettangolari: determinate le coordinate rettangolari (x, y) per il punto P (12, $\frac{4}{5}\pi$) in coordinate polari.

(Operazione) RADIAN [ENTER] (Specifica "radianti" come unità angolare)

REC (12, (4/5 ∗ π))

[ENTER] (x) **−9.708203933**

($x \fallingdotseq -9.7$)

Z [ENTER] (y) **7.053423028**

($y \fallingdotseq 7.1$)

* **I valori di y ed x vengono trasferiti rispettivamente alle variabili Z ed Y.**

Nota: Per la conversione delle coordinate, il risultato della conversione viene trasferito alle variabili Z ed Y. Pertanto, verrà annullato il valore precedente di Z ed Y.

— Riferimento —

Le equazioni costituite da operatori logici (=, >, <, >=, <=, <>) possono assumere i valori elencati nella tabella seguente:

x ed y rappresentano dei valori numerici.

= *	1 se $x = y$ 0 se $x \neq y$	>=	1 se $x \geqq y$ 0 se $x < y$
>	1 se $x > y$ 0 se $x \leqq y$	<=	1 se $x \leqq y$ 0 se $x > y$
<	1 se $x < y$ 0 se $x \geqq y$	<>	1 se $x \neq y$ 0 se $x = y$ ("<>" significa "≠".)

* Se per esempio, in un'equazione logica si usa "A = valore numerico" o "B = formula", il computer non la tratterà come equazione logica bensì come istruzione di assegnazione di un valore a una variabile. Quando usate un segno di uguale (=) in un'equazione logica, usatelo nella forma di "valore numerico = A" o "formula = B", con l'eccezione delle espressioni condizionali usate nelle istruzioni IF.

Funzione calcolo diretto

Nel calcolo manuale descritto sino ad ora, abbiamo sempre usato il tasto [ENTER] per porre termine ad una formula ed ottenere il risultato dell'operazione. Tuttavia, potete usare direttamente le funzioni del **computer** con il tasto di funzione desiderato (senza usare il tasto [ENTER]) quando il dato numerico, da operare, è sul visore.

(Esempio) Determinate sin 30° e 8!.

(Operazione) DEGREE [ENTER]

[C·CE] 30 — 30_

[sin] — 0.5

(Operazione) [C·CE] 8 — 8_

[SHIFT] [n!] — 40320.

(Esempio) Per $\tan^{-1}\frac{5}{12}$, controllate dapprima il risultato di $\frac{5}{12}$, quindi determinate $\tan^{-1}\frac{5}{12}$.

(Operazione) DEGREE [ENTER]

5/12 [ENTER] — 4.166666667E−01

[SHIFT] [tan⁻¹] — 22.61986495

Si deve tuttavia notare che questo modo di calcolo "diretto" non è possibile per le funzioni che richiedono l'immissione di più di un valore numerico (funzioni binomiali) quali la potenza, la potenza di radice o la conversione delle coordinate.
La funzione calcolo diretto non vale per le formule:

(Esempio) [C·CE] 5∗4 → 5∗4 _

[log] → 5∗4LOG _

La funzione calcolo diretto vale soltanto per i valori numerici. Pertanto, se vengono immessi i numeri esadecimali A ~ F per la conversione da esadecimale in decimale, la funzione calcolo diretto sarà inoperativa. In tal caso, usate il normale calcolo manuale con il tasto [ENTER] .

- Dopo aver eseguito un calcolo diretto, la funzione di richiamo non è operativa. Premendo i tasti [◄] o [►] si ottiene soltanto la visualizzazione del cursore.

Ordine di priorità nei calcoli manuali

Nel modo BASIC, potete inserire le formule nella stessa maniera in cui sono scritte, incluse le parentesi o le funzioni. Il computer gestirà da solo l'ordine di priorità nel calcolo ed il trattamento dei risultati intermedi.

L'ordine interno di priorità nei calcoli è il seguente:

1) Variabili o π.
2) Funzione (sin, cos, etc.)
3) Potenza (^) o potenza di radice (ROT)
4) Segno (+, –)
5) Moltiplicazione o divisione (*, /)
6) Addizione o sottrazione (+, –)
7) Confronto di grandezza (>, >=, <, <=, <>)
8) Operazioni logiche (AND, OR, NOT ed XOR)

Note:
* Se in una formula vengono usate le parentesi, l'operazione tra parentesi ha la massima priorità.
* Le funzioni composite vengono risolte da destra verso sinistra (sin $\cos^{-1}$ 0.6).
* Le potenze a catena (3^{4^2} o 3 ^4^2) o le potenze di radice vengono risolte da destra verso sinistra.
* Per i punti 3) e 4), l'ultimo dato ha una priorità superiore.

(Per esempio) $-2 \wedge 4 \rightarrow -(2^4)$

$3 \wedge -2 \rightarrow 3^{-2}$

CAPITOLO 4
CONCETTI E TERMINI DEL BASIC

In questo capitolo esamineremo alcuni concetti e termini del linguaggio BASIC.

Le stringhe costanti

Oltre ai numeri il **computer** Vi mette a disposizione oltre a tutte le lettere dell'alfabeto anche dei simboli speciali. Pertanto i numeri, le lettere ed i simboli, chiamati caratteri, a disposizione sul **computer** sono:

1 2 3 4 5 6 7 8 9 Ø
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
! " # $ % & ( ) * + , – . / : ; < = > ? @ √ π ^

Nel linguaggio Basic, un insieme di caratteri è chiamato **stringa** e pertanto dobbiamo assuefarci a questo particolare nome che spesso incontreremo in questo manuale. La stringa ha varie funzioni, sia descrittive che operative. Per permettere al **computer** di riconoscere una stringa da un comando in Basic, si usa racchiudere la parola tra virgolette (").
Ecco di seguito degli esempi di stringhe:

"CIAO"
"ARRIVEDERCI"
"SHARP"

I seguenti esempi sono invece sbagliati:

"COMPUTER	Mancano le virgolette di fine stringa. Si causa errore.
"C"E"	Non si possono usare le virgolette all'interno di una stringa.

Numeri esadecimali

Il sistema decimale è soltanto uno dei molti differenti sistemi per rappresentare dei numeri. Ai giorni nostri, con l'avvento del computer, è diventato sempre più popolare il sistema esadecimale. Questo sistema si basa sul numero 16 anziché il 1Ø. Per scrivere un numero esadecimale si usano sia le cifre da Ø a 9 che 6 caratteri: A, B, C, D, E e F, i quali corrispondono ai numeri 1Ø, 11, 12, 13, 14 e 15 (la prima cifra è lo zero e la 16ª è il 15).
Quando volete che il **computer** operi con un numero esadecimale fatelo precedere dal segno '&'.

&A = 1Ø
&1Ø = 16
&1ØØ = 256
&FFFF = 65535

Quelli di Voi che hanno una certa esperienza coi computer noteranno che l'ultimo numero (65535) è lo stesso che si colloca nello speciale gruppo di numeri visti nel paragrafo precedente. Per la programmazione o l'uso del **computer**, i numeri esadecimali non sono di fondamentale importanza; risultano utili per speciali applicazioni.

Le variabili

I computer sono costituiti da molte piccole aree di memoria chiamate bytes. Si può immaginare un byte come un singolo carattere. Per esempio, la parola byte richiede quattro byte di memoria in quanto è costituita da quattro caratteri. Per vedere quanti caratteri sono disponibili, basta battere MEM [ENTER] . Il numero visualizzato è il numero di byte disponibili per scrivere i programmi. Questa tecnica va benissimo con le parole, ma è molto inefficiente quando cercate di memorizzare dei numeri. Per questo motivo, i numeri vengono memorizzati in modo codificato. Grazie a questa tecnica di codificazione, il computer può memorizzare grossi numeri in soli 8 byte. Il numero più grande che può essere memorizzato è +9.999999999 E + 99. Il numero più piccolo è +1.E−99. In questo modo si ottiene una gamma di tutto rispetto in cui scegliere. Tuttavia, se il risultato di un calcolo supera questi limiti, il computer ve lo comunica con una segnalazione di errore sullo schermo. Questa segnalazione è costituita da una piccola E nell'angolo superiore di destra dello schermo. Per vederlo comparire ora, provate a battere:

9 [EXP] 99 ∗ 9 [ENTER]

Per ottenere che il computer torni a funzionare correttamente, basta premere il tasto [C·CE] . Come fare però a memorizzare tutte queste informazioni? Di fatto è molto semplice. Il computer ama usare dei nomi per i diversi dati. Memorizziamo nel computer il numero 556. Potete chiamare questo numero in qualsiasi modo, ma per questo esercizio, usiamo la lettera R. L'istruzione LET, può essere usata per dire al computer di assegnare un valore ad un nome di variabile ma soltanto in un'istruzione di programma. Tuttavia, il comando LET non è necessario, per cui non lo useremo molto spesso. Battete ora R = 556 e premete [ENTER]. Il computer ha ora associato il valore 556 alla lettera R. Queste lettere che vengono usate per memorizzare le informazioni vengono chiamate **variabili**. Per vedere il contenuto della variabile R, premete il tasto [C·CE] , il tasto R ed il tasto [ENTER]. Il computer risponde visualizzando il valorè 556 alla destra dello schermo. Questa funzione può diventare estremamente utile quando scrivete i programmi e le formule.

Usiamo ora la variabile R in una formula semplice. In questa formula, la variabile R indica il raggio di un cerchio di cui vogliamo trovare l'area. La formula dell'area del cerchio è: $A = \pi * R^2$. Battete R [SHIFT] [^] 2 [∗] [SHIFT] [π] [ENTER] . Il risultato è 971179.3866. Questa tecnica dell'uso delle variabili nelle equazioni diventerà più comprensibile quando cominceremo a scrivere i programmi.

Sino ad ora abbiamo parlato soltanto di variabili **numeriche**. E per quanto concerne la memorizzazione dei caratteri alfabetici? Bene, il concetto è uguale, ma per consentire al computer di riconoscere la differenza tra i due tipi di variabili, aggiungete un $ al nome della variabile. Per esempio, memorizzate la parola byte nella variabile B$. Avete notato il $ dopo la B? Questo dice al computer che il contenuto della lettera B è alfabetico, cioé una **stringa** di dati.

Provate a battere B [SHIFT] [$ R] = [SHIFT] [" W] B YTE [SHIFT] [" W] [ENTER] . Il valore BYTE è ora memorizzato nella variabile B$. Per esserne certi, battete B [SHIFT] [$ R] [ENTER] . Lo schermo visualizza BYTE. Questa volta la visualizzazione è alla sinistra del visore invece che a destra.

Le variabili gestite dal **computer** sono suddivise come segue:

- Variabili
  - Variabili numeriche
    - Variabili numeriche fisse (A–Z)
    - Variabili numeriche semplici (AB, C1, ecc.)
    - Matrici numeriche variabili
  - Variabili stringa
    - Variabili di caratteri fisse (A$–Z$)
    - Variabili di caratteri semplici (BB$, C2$, ecc.)
    - Matrici di caratteri variabili

Variabili fisse

La prima sezione, la **variabile fissa,** viene sempre usata dal computer per memorizzare i dati. Questa variabile può essere considerata uno spazio variabile preassegnato. In altre parole, indipendentemente da quanta memoria usa il vostro programma, disporrete sempre di almeno 26 variabili tra le quali scegliere per memorizzare i dati. Questi dati possono essere di due tipi: NUMERICI oppure STRINGA (caratteri alfabetici). Le posizioni fisse di memoria sono lunghe otto byte e possono essere usate per un solo tipo di dato per volta. A titolo illustrativo, battete l'esempio seguente:

A = 123 [ENTER]
A$ [ENTER]

Viene visualizzato il messaggio:

ERROR 9

Ciò significa che avete immesso dei dati numerici nell'area di memoria chiamata A e quindi avete detto al computer di rivisualizzare le informazioni come STRINGA. Questo mette in confusione il computer, pertanto vi comunica che vi è una condizione di errore. Premete il tasto [C-CE] per annullare la condizione di errore. Provate ora l'esempio seguente:

A$ = "ABC" [ENTER]
A [ENTER]

Nuovamente il computer è confuso e presenta il messaggio ERROR 9. Guardate la figura seguente e notate che il nome della variabile A ha la stessa area di memoria del nome della variabile A$, e che B è uguale a B$, e così via per tutte le lettere dell'alfabeto.

Figura:

```
A = A$ = A(1)  = A$(1)
B = B$ = A(2)  = A$(2)
C = C$ = A(3)  = A$(3)
D = D$ = A(4)  = A$(4)
E = E$ = A(5)  = A$(5)
F = F$ = A(6)  = A$(6)
G = G$ = A(7)  = A$(7)
H = H$ = A(8)  = A$(8)
I = I$ = A(9)  = A$(9)
J = J$ = A(10) = A$(10)
K = K$ = A(11) = A$(11)
L = L$ = A(12) = A$(12)
M = M$ = A(13) = A$(13)
N = N$ = A(14) = A$(14)
O = O$ = A(15) = A$(15)
P = P$ = A(16) = A$(16)
Q = Q$ = A(17) = A$(17)
R = R$ = A(18) = A$(18)
S = S$ = A(19) = A$(19)
T = T$ = A(20) = A$(20)
U = U$ = A(21) = A$(21)
V = V$ = A(22) = A$(22)
W = W$ = A(23) = A$(23)
X = X$ = A(24) = A$(24)
Y = Y$ = A(25) = A$(25)
Z = Z$ = A(26) = A$(26)
```

Variabili semplici

I nomi delle variabili semplici sono specificati da due (o più) caratteri alfanumerici, quali AA o B1. Diversamente dalle variabili fisse, le variabili semplici non hanno alcuna area preassegnata in memoria. L'area per le variabili semplici viene riservata automaticamente (nell'ambito dell'area dati/programma) quando viene usata per la prima volta.

Poiché vengono definite delle aree di memoria separate per le variabili numeriche semplici e per le variabili alfabetiche semplici anche se queste hanno lo stesso nome, è possibile usare, per esempio, contemporaneamente delle variabili quali AB e AB$.
Per i nomi delle variabili semplici è possibile usare dei caratteri alfanumerici, ma il primo carattere del nome della variabile deve sempre essere un carattere alfabetico. Se per definire un nome di variabile vengono usati più di due caratteri, sono significativi soltanto i primi due caratteri.

Nota:
- I nomi delle funzioni o delle istruzioni BASIC usati dal **computer** non possono essere utilizzati per i nomi delle variabili.
  (Esempio) PI, IF, TO, ON, SIN, etc.
- Ogni variabile semplice di carattere può contenere un massimo di 16 caratteri o simboli.

Matrici di variabili

Per alcuni usi è spesso molto utile poter lavorare con dei numeri organizzati in gruppi: guali potrebbero essere degli elenchi di punteggi, delle tavole di percentuali, etc. In BASIC questi gruppi sono chiamati matrici. Una matrice può avere una dimensione e pertanto presentarsi come una lista oppure essere bidimensionale come una tavola. Le matrici vengono indicate nella stessa maniera utilizzata per la variabili, la sola differenza è che la lettera viene seguita da un numero tra parentesi, nel caso di matrice ad una dimensione o da due numeri separati da una virgola, per le matrici bidimensionali.

Per definire una matrice, si usa l'istruzione DIM (abbreviazione di dimensione). Le matrici devono essere sempre "dichiarate" (definite) prima di poter essere usate. (Non come le variabili che abbiamo usato sinora). La forma dell'istruzione numerica DIM è:

DIM nome della variabile numerica (dimensione)

in cui:

il nome di variabile numerica è un nome di variabile conforme alle regole normali dei nomi di variabili numeriche precedentemente trattati.

Dimensione è il numero di posizioni di memoria e deve essere un numero compreso tra 0 e 255. Notate che quando specificate un numero per la dimensione, ottenete una posizione in più di quella specificata.

Esempi di istruzioni ammesse di dimensionamento delle matrici numeriche sono:

```
DIM X (5)
DIM AA (24)
DIM Q5 (0)
```

La prima istruzione crea una matrice X con 6 posizioni di memorizzazione. La seconda istruzione crea una matrice AA con 25 posizioni. La terza istruzione crea una matrice con una posizione ed è di fatto piuttosto inutile in quanto (quanto meno per i numeri,) è la stessa cosa che dichiarare una variabile numerica con un unico valore.

E' importante sapere che una matrice di variabili X ed una variabile X sono separate e distinte per la SHARP. La prima X indica una serie di locazioni di memorizzazione numeriche, mentre la seconda indica un'unica e diversa locazione.

Ora che sapete come creare le matrici, vi chiederete come si fa a fare riferimento a ciascuna posizione di memorizzazione. Poiché l'intero gruppo ha un solo nome, possiamo fare riferimento ad una singola posizione (chiamata "elemento") facendo seguire al nome del gruppo un numero tra parentesi. Questo numero viene chiamato "suffisso". Per esempio, per memorizzare il numero 8 nel quinto elemento della nostra matrice X (precedentemente dichiarata) scriviamo:

```
X(4) = 8
```

Se vi sconcerta il fatto di aver usato il numero 4 per indicare il quinto elemento, ricordate che la numerazione degli elementi comincia da zero e continua attraverso il numero della dimensione dichiarato nell'istruzione DIM.

Il potere reale delle matrici consiste nella capacità di usare un'espressione o un nome di variabile come indice.

Per dichiarare una matrice di caratteri, si usa una forma leggermente diversa dell'istruzione DIM:

<u>DIM</u> nome della variabile di caratteri (dimensione) * lunghezza

in cui:

<u>il nome della variabile</u> di caratteri è un nome di variabile conforme alle norme relative alle variabili alfabetiche normali precedentemente trattate.

<u>Dimensione</u> è il numero di posizioni di memorizzazione e deve essere compreso nella gamma tra 0 e 255. Notate che quando specificate un numero, ottenete una posizione in più di quella specificata.

*lunghezza è facoltativo. Se viene usato, specifica la lunghezza di ciascuna delle stringhe che costituiscono la matrice. La lunghezza è data da un numero compreso tra 1 e 8Ø. Se non viene usata questa clausola, le stringhe avranno la lunghezza assunta automaticamente di 16 caratteri.

Diamo degli esempi di dichiarazioni di matrici di carratteri ammesse:

```
DIM X$ (4)
DIM NM$ (10) * 10
DIM IN$ (1) * 80
DIM R$ (0) * 26
```

Il primo esempio crea una matrice di 5 stringhe, ciascuna delle quali può essere costituita da 16 caratteri. La seconda istruzione DIM dichiara una matrice NM con 11 stringhe fino a 10 caratteri ciascuna. La definizione di stringhe di lunghezza inferiore a quella assunta automaticamente, aiuta a conservare dello spazio di memoria. Il terzo esempio dichiara una matrice di due elementi costituita da stringhe di 80 caratteri e l'ultimo esempio dichiara un'unica stringa di 26 caratteri.

Oltre alle matrici semplici appena esaminate, il **computer** consente l'uso delle matrici "bidimensionali". Per analogia, una matrice unidimensionale è una lista di dati disposta su di una singola colonna. Una matrice bidimensionale è una tabella di dati con delle righe e delle colonne. La matrice bidimensionale viene dichiarata mediante l'istruzione:

DIM nome di variabile numerica (righe, colonne)

oppure

DIM nome di variabile alfabetica (righe, colonne) * lunghezza

in cui:

Righe specifica il numero di righe della matrice. Questo numero deve essere compreso tra 0 e 255. Notate che quando specificate il numero di righe, ottenete una riga in più del numero specificato.

Colonne specifica il numero di colonne della matrice. Questo numero deve essere compreso tra 0 e 255. Notate che quando specificate il numero delle colonne, ottenete una colonna in più del numero specificato.

Il diagramma seguente illustra le posizioni di memorizzazione che risultano dalla dichiarazione DIM T (2, 3) ed i suffissi (ora costituiti da due numeri) che si riferiscono a ciascuna posizione di memorizzazione:

	Colonna 1	Colonna 2	Colonna 3	Colonna 4
riga 0	T (0, 0)	T (0, 1)	T (0, 2)	T (0, 3)
riga 1	T (1, 0)	T (1, 1)	T (1, 2)	T (1, 3)
riga 2	T (2, 0)	T (2, 1)	T (2, 2)	T (2, 3)

NOTA: Le matrici bidimensionali possono occupare rapidamente lo spazio di memoria. Per esempio, una matrice con 25 righe e 35 colonne usa 875 posizioni di memorizzazione!

Le tabella seguente mostra il numero di byte usato per definire ogni variabile ed il numero usato da ciascuna istruzione di programma.

Variabile	Nome di variabile	Dato	
Variabile numerica	7 byte	8 byte	
Variabile stringa	7 byte	Matrice variabile	Numero specificato di byte
		Variabile semplice (variabile di due caratteri)	16 byte

* Per esempio, se si specifica DIM Z$ (2, 3) * 10, vengono riservate 12 variabili, ciascuna in grado di memorizzare 10 caratteri. Questo richiede 7 byte (nome della variabile) + 10 byte (numero dei caratteri) x 12 = 127 byte.

Elemento	Numero di riga	Istruzione e funzione	Diversi, ENTER
Numero di byte usati	3 byte	1 byte o 2 byte	1 byte

Variabili in forma di A ( )

Benché venga riservata un'area dati nella memoria del computer per le variabili fisse, questa può essere usata anche per definire le variabili con suffissi che hanno la stessa forma delle matrici di variabili.

Sono disponibili 26 nomi di variabili fisse: cioé A–Z (A$–Z$). A ciascuno di questi nomi è possibile apporre un suffisso costituito dai numeri 1–26, come per esempio A(1)–A(26) o A$(1)–A$(26). Ciò significa che la variabile A(1) può essere usata in luogo della variabile A, A(2) in luogo di B, A(3) in luogo di C, e così via.

Tuttavia, se una matrice chiamata A o A$ è già stata definita dall'istruzione DIM, non è possibile definire le variabili con suffissi A. Per esempio, se una matrice A è definita da DIM A(5), la posizione di A(0)–A(5) viene riservata nell'area programma dati. Pertanto, se specificate la variabile A(2), questa non si riferisce alla variabile fissa B, bensì si riferisce alla matrice di variabile A(2) definita nell'area programma/dati.

Se specificate A(9), si verificherà un errore poiché A(9) è al di fuori della gamma di dimensioni specificata dall'istruzione DIM A(5).

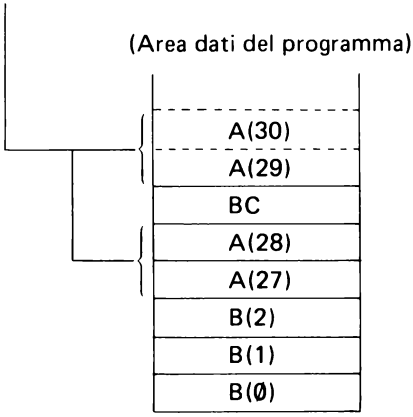
Viceversa, se le variabili con i suffissi sono già state definite nella forma A( ), non è possibile definire le matrici A o A$ usando l'istruzione DIM, a meno che la definizione delle variabili con i suffissi venga annullata con l'istruzione CLEAR.

* Uso di un numero di suffissi superiore a 26:
  Se si usano più di 26 suffissi per le variabili A( ) quando la matrice A non è definita da un'istruzione DIM, vengono riservate delle posizioni corrispondenti nell'area programma dati per queste variabili fisse. Per esempio, se eseguite A (35) = 5, nell'area programma dati verranno riservate le posizioni per le variabili A (27) – A (35).
  Le variabili con suffissi superiori a 26 vengono trattate come matrici di variabili, sono però soggette alle seguenti limitazioni speciali:

(1) Le posizioni di una matrice con lo stesso nome devono essere contigue nell'area programma/dati, diversamente si verifica un errore.

```
10 DIM B(2)
20 A(28) = 5
30 BC = 12
40 A(30) = 9
```

Se viene eseguito questo programma, la matrice chiamata "A" non è definita in due segmenti consecutivi nell'area programma dati e, pertanto, nella riga 40 si avrà un errore.



(2) Le matrici di variabili numeriche e le matrici di variabili alfabetiche con lo stesso suffisso non possono essere definite contemporaneamente. Per esempio, A(30) e A$(30) non possono essere definite contemporaneamente, in quanto usano la stessa posizione nell'area programma/dati.

(3) Non è possibile definire due matrici dimensionali, come non è possibile specificare la lunghezza delle stringhe di caratteri che devono essere contenute nelle matrici di variabili alfabetiche. Per esempio, la lunghezza di una stringa di caratteri che può essere contenuta nella matrice di variabili alfabetiche A$( ) è limitata a 7 caratteri o meno.

(4) Non è possibile definire delle variabili con suffisso zero (0). Se si definisce A(0) o A$(0), si verifica un errore.

(5) Se si usano degli indici maggiori di 27, 7 byte sono usati per il nome della variabile, mentre 8 sono usati per la definizione di ogni singola variabile.

Espressioni

Un'espressione è una combinazione di variabili, costanti ed operatori matematici che danno un risultato. I calcoli che abbiamo eseguito nel capitolo 3 erano delle espressioni. Le espressioni sono una parte fondamentale dei programmi in BASIC, infatti un'espressione potrebbe essere una formula per calcolare i risultati di un'equazione oppure un test per determinare la relazione esistente tra due quantità o ancora un mezzo per dimensionare delle stringhe, etc.

Operatori numerici

Il **computer** ha cinque operatori numerici, gli stessi che avete usato nel capitolo 3 studiando l'uso del computer come calcolatrice, essi sono:

+	Addizione	/	Divisione
–	Sottrazione	^	Potenza
*	Moltiplicazione		

Un'espressione numerica è costruita nella stessa maniera utilizzata nel capitolo 3 per i calcoli composti. Le espressioni numeriche possono contenere qualsiasi combinazione significativa di costanti numeriche, variabili numeriche e di operatori numerici:

```
(A * B) ^ 2
A (2, 3) + A (3, 4) + 5.Ø – C
(A/B) * (C+D)
```

Espressioni stringa

Le espressioni stringa sono simili a quelle numeriche con la differenza che in quelle stringa vi è un unico operatore aritmetico ammesso che è il segno +. Questo segno consente di sommare due o più stringhe, ovvero il contenuto della seconda stringa viene aggiunto alla fine della prima stringa, dando cosí come risultato una stringa più lunga. Dovete fare attenzione quando fate delle concatenazioni di stringa ed altre operazioni sulle stringhe in quanto lo spazio di lavoro usato del **computer** per i calcoli con le stringhe è limitato a soli 79 caratteri.

Nota: Le stringhe ed i numeri non possono essere combinati nella stessa espressione a meno che si usi una delle funzioni che converte una stringa in un numero o viceversa:

```
"15" + 1Ø      è illecito
"15" + "1Ø"    è "151Ø" e non "25"
```

Operatori logici/espressioni di relazione

Gli operatori logici permettono dei paragoni tra due grandezze determinando se la relazione è vera o falsa, e sono:

>	Maggiore di
>=	Maggiore uguale di
=	Uguale
<>	Differente
<=	Minore uguale di
<	Minore di

I seguenti sono corretti esempi di relazioni:

```
A < B
C(1, 2) >= 5
D(3) <> 8
```

Se A fosse 1Ø e B fosse 12, C(1, 2) fosse 6 e D(3) fosse 9, tutti questi paragoni sarebbero veri.

Oltre alle grandezze numeriche, i paragoni si possono effettuare anche sulle stringhe. In questo caso due stringhe vengono comparate, l'una all'altra, carattere per carattere, partendo dal primo ed in base al codice ASCII. (Vedere appendice B per i valori del codice ASCII).

Se una stringa è più corta dell'altra, verrà usato uno Ø o un NUL per ogni posizione mancante. Qui di seguito alcuni esempi di paragoni:

"ABCDEF" = "ABCDEF"
"ABCDEF" <> "ABCDE"
"ABCDEF" > "ABCDE"

Questi paragoni sia numerici che stringhe, vengono verificati se veri o falsi. Il **computer** rappresenta il vero con un 1; il falso con un Ø.
La pratica che acquisirete nella programmazione, vi suggerirà le utilizzazioni di questi operatori.

Espressioni logiche

Le espressioni logiche sono relazioni che si servono degli operatori AND, OR, XOR e NOT. AND, OR e NOT sono usate per collegare due espressioni ; il valore delle espressioni combinate è rappresentato nelle seguenti tabelle.

A AND B

		Valore di A	
		Vero	Falso
Valore di B	Vero	Vero	Falso
	Falso	Falso	Falso

A OR B

		Valore di A	
		Vero	Falso
Valore di B	Vero	Vero	Vero
	Falso	Vero	Falso

A XNOR B

		Valore di A	
		Vero	Falso
Valore di B	Vero	Falso	Vero
	Falso	Vero	Falso

Nota: Il valore di A e di B deve essere 0 (falso) o 1 (vero).

L'istruzione XOR non può essere usata in combinazione con le istruzioni AND ed OR all'interno di una espressione. Per eseguire l'espressione D=(A XOR B) AND C, per esempio, è necessario dividerla in due parti eseguite separatamente; D=A XOR B e D=D AND C.

- I numeri decimali possono essere espressi nel codice binario di 16 bits come segue:

NOTAZIONE DECIMALE	NOTAZIONE BINARIA 16 BITS
32767	0111111111111111
⋮	
3	0000000000000011
2	0000000000000010
1	0000000000000001
0	0000000000000000
−1	1111111111111111
−2	1111111111111110
−3	1111111111111101
⋮	⋮
−32768	1000000000000000

Il negativo (NOT) di un numero binario 0000000000000001 è calcolato come segue:

NOT	0000000000000001
(negativo) ⟶	1111111111111110

In questo modo 1 diventa 0 e 0 diventa 1 per ogni bit, operazione chiamata "prendere il negativo NOT".
Ecco cosa succede quando 1 e NOT 1 sono sommati:

```
     0000000000000001  (1)
+)   1111111111111110  (NOT 1)   ............Complemento di uno
     ------------------------------------------
     1111111111111111  (−1)      ............Complemento di due
```

In questo modo tutti i bit diventano 1. Pertanto , secondo quanto visto, il numero secondo il sistema decimale diventa −1 ovvero 1 + NOT 1 = −1.
La relazione tra un valore numerico X ed il suo negativo (NOT X) è:

X + NOT X = −1

Che risulta dall'equazione di NOT X = − X − 1
ovvero NOT X = −(X + 1)
Dall'equazione vengono ricavati i risultati:

NOT 0 = −1
NOT −1 = 0
NOT −2 = 1

Con questi operatori si possono combinare più di due espressioni. Dovete fare attenzione ad usare le parentesi per rendere chiaro il paragone.

```
(A<9) AND (B>5)
(A>=10) AND NOT (A>20)
(C=5) OR (C=6) OR (C=7)
(X>=50) XOR (X<70)
```

Il **computer** considera gli operatori logici quali "Bitsaggi" di funzioni logiche aventi 16 bit (vedere operatori logici vero-falso). Nelle normali operazioni questo non è significativo poiché il semplice 1 o Ø (vero, falso) che risulta da un paragone, usa solo un bit. Se applicate ad un operatore logico un valore differente di Ø o 1, esso lavorerà indipendentemente su ogni bit.
Per esempio se A è 17 e B è 22 (A OR B) è 23:

```
17 OR 22 è 10001 .. 17
           10110 .. 22   } Operazione OR
           ------------
           10111 .. 23 in numeri decimali
```

17 e 22 sono dapprima convertiti in numeri binari. Quindi, per ciascuna cifra, l'operatore logico 1 viene lasciato se ambedue i bit sono 1. Altrimenti viene lasciato l'operatore logico 0.
Per esempio, se A è 41 e B è 27, (A XOR B) è 50)

```
41 XOR 27 è 101001 .. 41
            011011 .. 27   } Operazione XOR
            ------------
            110010 .. 50 in numeri decimali
```

41 e 27 sono dapprima convertiti in numeri binari. Quindi, per ciascuna cifra, l'operatore logico 0 viene lasciato se ambedue i bit sono 1 o 0.a

Parentesi e precedenze

Quando valuta delle espressioni complesse, il **computer** segue una serie predeterminata di priorità che determinano la sequenza in cui i calcoli vengono svolti. Vediamo l'esempio seguente:

```
5 + 2 * 3  potrebbe essere
5 + 2 = 7                    oppure        2 * 3 = 6
7 * 3 = 21                                 6 + 5 = 11
```

Le regole esatte della "precedenza degli operatori" vengono date nell'appendice D.

Per evitare di dover ricordare tutte queste regole e per rendere più chiaro il vostro programma, usate sempre le parentesi per determinare la sequenza della valutazione. L'esempio precedente viene chiarito scrivendo:

```
(5 + 2) * 3                  oppure        5 + (2 * 3)
```

Modo RUN

In generale, le espressioni che abbiamo visto fino a questo momento, possono essere calcolate sia utilizzando il **computer** come calcolatrice, sia come parti di programma in BASIC.
Utilizzando il **computer** come calcolatrice, nel modo RUN, l'espressione viene calcolata immediatamente ed il risultato viene visualizzato. Per esempio:

Premere	Visore
(5 > 3) AND (2 < 6)	1.

1 significa che la relazione è vera.

Funzioni

Le **funzioni** sono delle componenti speciali del linguaggio BASIC che prendono un valore e lo trasformano in un altro. Queste funzioni agiscono come delle variabili i cui valori sono determinati da altre variabili o espressioni. Per esempio la funzione ABS determina il valore assoluto dell'argomento:

ABS (–5) è 5
ABS (6) è 6

La funzione LOG calcola il logaritmo in base 10 dell'argomento.

LOG (100) è 2
LOG (1000) è 3

Una funzione può essere usata in qualsiasi posizione in cui può essere usata una variabile. Molte funzioni non richiedono l'uso delle parentesi:

LOG 100 è identico a LOG (100)

Dovete usare le parentesi per le funzioni che hanno più di un argomento. Usando le parentesi, sareste sicuri di fare dei programmi chiari.
Al capitolo 8 vengono descritte tutte le funzioni possibili sul **computer**.

CAPITOLO 5
PROGRAMMAZIONE

Nei capitoli precedenti abbiamo esaminato alcuni concetti e termini della programmazione in BASIC.
In questo capitolo userete queste informazioni per creare dei programmi sul **computer**. Lasciateci però ripetere che questo non è un manuale che insegna la programmazione in BASIC, ma soltanto un mezzo per familiarizzarvi con BASIC del vostro **computer**.

Programmi

Un programma è costituito da un insieme di istruzioni che vengono date al computer, ricordatevi che il **computer** è solo una macchina e come tale eseguirà quello che voi specificherete.

Voi, come programmatore, siete responsabile dell'uso corretto delle istruzioni.

Dichiarazioni (statement) in BASIC

Il **computer** interpreta le istruzioni ricevute secondo un formato predeterminato. Questo formato è chiamato statement (tradotto in italiano con: dichiarazione). Le dichiarazioni BASIC vengono immesse sempre nello stesso schema. Le dichiarazioni devono cominciare con un numero di riga:

```
10: PRINT "CIAO"
20: END
30:  :
```

Numero di riga

Ogni riga di un programma deve avere un numero unico che può essere un qualsiasi numero intero compreso tra 1 e 65279. I numeri di riga sono il riferimento per il computer. Questi numeri dicono al **computer** l'ordine in cui il programma va eseguito. Non è necessario immettere le righe in ordine progressivo (sebbene, nel caso siate un principiante, sia probabilmente più chiaro procedere in tal modo). Il computer comincia sempre l'elaborazione dal numero di riga più basso per continuare progressivamente lungo le righe di un programma in ordine ascendente.

In fase di programmazione, è bene numerare le righe lasciando degli intervalli tra due numeri (10, 20, 30, ... 10, 30, 50 etc). Questo vi consente di inserire delle righe ulteriori nel caso fosse necessario.
ATTENZIONE: Non usate lo stesso numero di riga in programmi diversi. Se usate lo stesso numero di riga, quando date il comando di memorizzazione della nuova riga, questa cancella la vecchia e si sostituisce ad essa.

I verbi in BASIC

Tutte le istruzioni in BASIC devono contenere dei **verbi**. I verbi dicono al computer quale azione deve compiere. Un verbo è sempre contenuto in un programma e come tale non viene elaborato immediatamente.

```
10: PRINT "CIAO"
20: END
30:   ⋮
```

Alcune istruzioni richiedono o consentono un **operando**:

```
10: PRINT "CIAO"
20: END
30:   ⋮
```

Gli operandi forniscono al computer le informazioni con le quali gli dicono su quali dati agirà il verbo. Alcuni verbi richiedono degli operandi mentre con altri verbi gli operandi sono facoltativi. Alcuni verbi non consentono l'uso degli operandi. (Vedere il capitolo 8 per l'elenco completo dei verbi BASIC e del loro uso nel **computer**.)

I comandi in BASIC

I comandi sono delle istruzioni che vengono date direttamente dalla tastiera e che ordinano al computer di eseguire determinate azioni nell'ambito del vostro programma o di impostare i modi che influenzano il modo di esecuzione dei programmi.

Diversamente dai verbi, i comandi hanno un effetto immediato. Infatti, non appena avete scritto la parola corrispondente al comando desiderato e premuto il tasto [ENTER], tale comando verrà immediatamente eseguito. I comandi non sono preceduti da un numero di riga:

RUN ordina l'esecuzione del programma in memoria.

NEW ordina di cancellare tutta la memoria.

RADIAN ordina il cambiamento del metodo di rappresentazione degli angoli.

Alcuni verbi possono essere usati anche come comandi. (Vedere il capitolo 8 per l'elenco completo dei comandi in BASIC e del loro uso sul **computer**.)

Modi di funzionamento

Il modo RUN viene utilizzato anche per eseguire i programmi da voi creati.

Il modo PRO viene utilizzato sia per immettere che per correggere i programmi.

Cominciamo a programmare il computer

Gli esercizi che avete fatto usando il **computer** come calcolatrice, vi avranno senz'altro familiarizzati con l'uso della tastiera. D'ora in avanti, quando vi illustreremo qualcosa da memorizzare, non indicheremo più la sequenza dei tasti. Ricordatevi di usare [SHIFT] per accedere ai simboli od alle funzioni poste sopra ai tasti e di porre termine ad ogni riga premendo il tasto [ENTER] .

Ora siete pronti per la programmazione.
Mettete l'interruttore di accensione in posizione ON e quindi premete il tasto BASIC due volte. Sul visore vedrete le seguenti informazioni iniziali.

La visualizzazione precedente indica che il computer è nel modo PROgramma. (Se l'etichetta CAL o RUN è accompagnata da una lineetta, premete il tasto BASIC uno o due volte.)
Date il comando NEW.

Premere	Visore
NEW [ENTER]	>

Il comando NEW cancella dalla memoria del **computer** tutti i programmi e tutti i dati. Il simbolo di pronto, riappare non appena premete il tasto [ENTER] .

Esempio N. 1 – Inseriamo ed elaboriamo un programma

AssicurateVi che il **computer** sia in posizione PRO, quindi impostate il seguente programma:

Premere	Visore
1Ø PRINT "CIAO"	1Ø: PRINT "CIAO"

Dopo aver premuto il tasto [ENTER] , il computer, automaticamente, inserisce due punti tra il numero di riga ed il verbo. Verificate sempre che le varie istruzioni siano scritte correttamente.
Ora premete il tasto [BASIC] per tornare in posizione RUN.

Premere	Visore
RUN [ENTER]	CIAO

Essendo questa l'unica riga del programma, il computer pone termine all'esecuzione a questo punto. Volendo far ripetere il messaggio premere [ENTER] per uscire dal programma e quindi nuovamente RUN.

Esempio N. 2 – Revisione del programma

Supponete di voler cambiare il messaggio che il Vostro computer visualizza, ovvero, volete correggere il Vostro programma, ovviamente, avendo una sola riga di programma sarebbe più facile riscriverla ex-novo, ma quando avrete programmi molto più complicati, la possibilità di correggerli, senza riscriverli, diventa molto utile. Rivediamo il programma che avete inserito.

Avete il **computer** ancora in posizione RUN? Se si, portatelo in PRO.

Dovete richiamare il programma per poterlo rivedere. Ricchiametelo perciò servendovi del tasto con la freccia rivolta verso l'alto (↑). Se il vostro programma è stato completamente eseguito, il tasto [↑] richiama l'ultima riga del programma. Se il programma contiene un errore, o se ne avete interrotto l'esecuzione con il tasto BREAK [BRK] , il tasto [↑] richiama la linea contenente l'errore o in corrispondenza della quale BREAK è stato premuto. Per rivedere il programma, scorretelo verso l'alto premendo il tasto con la freccia verso l'alto [↑] (richiamando la riga precedente) e verso il basso premendo il tasto con la freccia verso il basso [↓] (Visualizzando la riga successiva). Se mantenuti premuti, i tasti [↑] e [↓] scorrono verticalmente il programma, cioè si muoveranno di riga in riga verso l'alto o verso il basso.

Ricorderete che per spostare il cursore su di una riga potete usare il tasto (>) freccia verso destra) e quello (<) (freccia verso sinistra). Portate il cursore in corrispondenza del primo carattere che volete cambiare. con il tasto [►] .

Premere	Visore
[↑]	10: PRINT "CIAO"
[◄] [◄] [◄] [◄]	10 PRINT "CIAO"

Notate che il cursore si trova, ora, nella forma lampeggiante, forma che viene assunta quando si trova su di un carattere. Scrivete:

Premere	Visore
VADO"!	10 "VADO" ! _

Premendo il tasto [ENTER] alla fine della riga, e passando quindi al modo RUN premendo il tasto [BASIC], nel display appare quanto segue:

Premere	Visore
RUN [ENTER]	ERROR 1 IN 10

Questo è un nuovo messaggio d'errore. Non solo viene indicato il tipo d'errore (il nostro vecchio amico: l'errore di sintassi) ma viene indicata anche la riga dove è stato trovato.

Premete il tasto [C·CE] e ritornate al modo PRO. Dovete essere in PRO per poter correggere o modificare un programma. Usando il tasto [↑] richiamiamo l'ultima riga del programma.

Premere	Visore
[↑]	10: PRINT "VADO" !

Il cursore lampeggia nel punto nel quale è stato identificato l'errore. Nel capitolo 4 avevate visto che le stringhe devono essere contenute nelle virgolette. Il ! si trova al di fuori causando una situazione d'errore. Utilizzate il tasto DEL per eliminare il !.

Premere	Visore
[SHIFT] [DEL]	10 PRINT "VADO" _

Ora mettiamo il punto esclamativo nella sua giusta posizione.

Nella correzione di un programma, le funzioni DEL ed INS vengono utilizzate nella stessa maniera di quella utilizzata per correggere i calcoli (vedere capitolo 3). Pertanto, utilizzando il tasto [◄] posizionamo il cursore sul primo carattere che sarà il successivo a quello che vogliamo inserire.

Premere	Visore
[◄]	1Ø PRINT "VADO"

Premete il tasto INS. Uno spazio ⎵ così formato e lampeggiante, indica il punto dove si verrà a collocare il carattere che inseriremo.

Premere	Visore
[SHIFT] [INS]	1Ø PRINT "VADO ⎵ "

Inserite il punto esclamativo:

Premere	Visore
!	1Ø PRINT "VADO ! "

RicordateVi di premere [ENTER] affinché la riga corretta si sostituisca alla precedente.

Nota: Se volete cancellare un'intera riga di programma è sufficiente in posizione PRO scrivere il numero della riga e premere il tasto [ENTER].

Esempio N. 3 – L'uso delle variabili nel programma

Se non conoscete nel Basic l'uso delle variabili numeriche e di quelle stringa, Vi suggeriamo di rileggere il capitolo 4 ove si tratta di questi argomenti.

La buona conoscenza dell'uso delle variabili Vi permette di sfruttare pienamente la capacità del Vostro **computer**.

Ricorderete che per assegnare un numero ad una variabile numerica potete utilizzare qualsiasi lettera da A a Z:

A = 5

Per assegnare una stringa ad una variabile stringa si utilizza una lettera seguita dal segno del dollaro, da A$ e Z$. **Non usate la stessa lettera per la designazione di una variabile numerica e una di stringa.** Non dovete, nello stesso programma, avere la variabile A e A$ contemporaneamente (una annulla l'altra).

RicordateVi, inoltre, che una semplice variabile stringa può contenere al massimo 7 caratteri:

A$ = "TOTALE"

Il valore assegnato ad una variabile, può cambiare durante l'esecuzione del programma, assumendo i valori che vengono impostati, o quelli che vengono calcolati.

Una via per assegnare un valore ad una variabile è quello di utilizzare il verbo INPUT (INSERISCI). Nel seguente programma il valore di A$ cambierà in funzione della risposta che si dà alla domanda "PAROLA?".

Impostate il seguente programma:

```
10 INPUT "PAROLA?" ; A$
20 B = LEN (A$)
30 PRINT "PAROLA␣BI␣":B;"␣LET."
40 END
```

Questo simbolo significa spazio.

Prima di far eseguire questo programma, soffermiamoci su alcune nuove caratteristiche. La linea 30 supera i 24 caratteri che, come sapete, sono il massimo che il **computer** può visualizzare contemporaneamente. Pertanto, quando siete al 25° carattere, il **computer**, automaticamente sposta a sinistra di un carattere la riga facendola così scorrere orizzontalmente fino a contenere un massimo di 79 caratteri. Ovviamente i caratteri che scompaiono a sinistra non vengono cancellati.

Alla riga 40 troviamo l'istruzione END (FINE) per segnalare al computer che il programma è completato. E' buona norma terminare il programma con questo comando per evitare, quando avrete vari programmi in memoria, che il computer passi a quello successivo senza un Vostro esplicito ordine.

Poiché il Vostro programma comincia a diventare complesso, potrebbe nascere la necessità di rivederlo per controllarlo, per fare questo potete ricorrere al comando LIST. Questo comando agisce solo in PRO e visualizza il programma partendo dalla riga col numero più basso.

Proviamo ad effettuare questo listato:

Premere	Visore
LIST [ENTER]	10: INPUT "PAROLA

Usate il tasto [↑] e [↓] per far scorrere il programma, fino a quando lo avete passato tutto in rassegna. Per controllare una riga di programma che consta di più di 16 caratteri, sposate il cursore all'estrema destra del display e appariranno così gli altri caratteri.
Vediamo ora se il nostro programma funziona. Portiamo il computer in RUN.

Premere	Visore
RUN [ENTER]	PAROLA?_
AIUTO [ENTER]	PAROLA DI 5. LET
[ENTER]	>

Il Vostro programma è terminato. Naturalmente potreste ricominciare da capo reimpostando il RUN, oppure renderlo più divertente operando una modifica che ci permetterà di continuare il programma senza dover scrivere ogni volta RUN.

Ritornate in PRO e, utilizzando le freccie [↑] [↓] o il comando LIST, portateVi alla riga 4Ø.

Ora potete scrivere 4Ø e premere [ENTER] per cancellare questa riga oppure portare il cursore sulla E di END per inserire la nuova riga, come avevamo visto più sopra, effettuate la seguente correzione:

4Ø: GOTO 1Ø

Ora fate girare il programma modificato.

L'istruzione GOTO (VAI A . . .) fa fare al programma un certo salto e, nel nostro caso, questo salto lo riporta alla prima istruzione generando così un circolo chiuso. Poiché non vi è un limite alle volte che vogliamo ripetere questo giro, il computer continuerà all'infinito a girare su questo programma. Per fermare questo programma premete il tasto BREAK ([BRK]).

Quando fermate il programma col tasto [BRK] , sul visore appare la scritta BREAK IN. Se volete continuare, utilizzate il comando CONT, scrivendo CONT e premendo ENTER. Con il comando CONT il programma riparte dalla riga dove era stato effettuato il BREAK.

Esempio N. 4 – Programmi più complessi

Benché il computer comporti la funzione del fattoriale, in questa sezione useremo l'esempio del calcolo del fattoriale per spiegare delle procedure di programmazione più complesse.
Il programma che segue calcola il fattoriale di un numero (N!). Il programma comincia dal numero 1 e continua sino al limite da voi inserito. Inserite il seguente programma.

```
100 F = 1: WAIT 118
110 INPUT "LIMIT? "; L
120 FOR N = 1 TO L
130 F = F * N
140 PRINT N, F
150 NEXT N
160 END
```

Come potete vedere ci troviamo di fronte a parecchie nuove funzioni. Il verbo WAIT (attendere) alla riga 100 controlla il tempo per il quale i dati rimangono visualizzati sul visore prima che il programma continui. I numeri ed i loro fattoriali vengono visualizzati quando vengono calcolati. Il tempo per il quale rimangono sul visore viene determinato dall'istruzione WAIT in circa 2 secondi, invece di aspettare che premiate il tasto [ENTER].

Notate inoltre che alla riga 100 vi sono due istruzioni sulla stessa riga separate da due punti. E' possibile inserire quante istruzioni si vogliono in una sola riga, separandole con un due punti, purché si rispetti il limite massimo di 80 caratteri incluso il tasto [ENTER]. Le righe che contengono diverse istruzioni possono rendere però difficoltosa la lettura e la modifica di un programma, pertanto è buona pratica inserire più istruzioni in una sola riga soltanto quando queste istruzioni sono molto semplici o quando esiste un motivo particolare per volere che le istruzioni siano su di una sola riga.

Inoltre in questo programma abbiamo usato il verbo FOR alla riga 120 ed il verbo NEXT alla riga 150 per creare un anello (ciclo). Nell'esempio 3, avete creato un anello "infinito" che continuava a ripetere le istruzioni all'interno dell'anello sino a che non premevate il tasto [BRK]. I verbi FOR/NEXT lavorano solo insieme e servono per effettuare un calcolo ripetitivo per un determinato numero di volte. Infatti, contrariamente a quanto visto nell'esempio 3 dove avevamo creato un anello nel quale certe operazioni venivano effettuate un numero infinito di volte e per interrompere bisognava agire sul tasto [BRK], coi verbi FOR e NEXT si può decidere per quante volte quel certo calcolo debba essere ripetuto. Per far questo esiste un contatore che nel nostro esempio è rappresentato dalla variabile N ed il limite del conteggio della variabile L. Ogni volta che il programma passerà alla riga 150,

aggiungerà automaticamente un'unità al contenuto della variabile N e quando questa sarà maggiore di L, salterà automaticamente alla riga 160 uscendo dal ciclo FOR/NEXT e terminando il programma.

Potete usare qualsiasi variabile numerica in un anello FOR/NEXT. Non è neppure necessario cominciare a contare da 1 e potete aggiungere qualsiasi cifra ad ogni passo. Vedere il capitolo 8 per i dettagli.

Come avete notato, il nostro esempio comincia dalla riga 100. L'uso di differenti numeri di riga vi consente di memorizzare vari programmi contemporaneamente. Per eseguire (RUN) questo programma invece di quello della riga 10, scrivete:

RUN 100

Oltre ad eseguire i diversi programmi dando il loro numero di riga iniziale, potete dare un nome al programma con una lettera dell'alfabeto e farlo cominciare con il tasto DEF (vedere capitolo 6).

Noterete che quando il programma è in corso, appare la parola BUSY in alto a sinistra, per indicare che il **computer** sta elaborando. Provate questo programma per alcune volte cambiando i valori.

Permanenza dei programmi nella memoria del computer

Ricorderete che le impostazioni e le funzioni nonché i programmi rimangono in memoria nel computer anche quando il **computer** viene spento ed anche nel caso intervenga lo spegnimento automatico. Neppure l'uso dei tasti [BRK] , [C·CE] o [CA] cancella i programmi.

I programmi vengono persi dalla memoria soltanto quando vengono effettuate le seguenti operazioni:

* Impostate il comando NEW prima di cominciare la programmazione.
* Inizializzate il computer usando il tasto ALL RESET.
* Create un nuovo programma usando gli stessi numeri di riga del programma già in memoria.
* Cambiate le batterie.

Questa breve introduzione alla programmazione sul **computer** è stata fatta per illustrarvi le fantastiche possibilità di questo computer.

CAPITOLO 6
SCORCIATOIE

Il **computer** include numerose caratteristiche per rendere più facile la programmazione riducendo la quantità di tasti che si devono premere per le funzioni ripetitive.

Una di queste è l'uso della forma abbreviata per inserire comandi e verbi, illustrati al capitolo 8, le altre sono: l'uso del tasto DEF (Defined=Definito).

Il tasto DEF ed i programmi etichettati

Spesso Vi capiterà di memorizzare differenti programmi contemporaneamente (ricordateVi che ognuno dovrà avere le sue righe di programma). Naturalmente cominciate a far elaborare un programma con il comando RUN o con GOTO, però Vi dovete ricordare il numero della riga da dove comincia il programma che desiderate elaborare. Esiste anche un sistema più semplice, potete, infatti etichettare ogni programma con una lettera e farlo girare usando [DEF].

Nota: Mettete un'etichetta in corrispondenza di ciascun programma che volete contrassegnare. L'etichetta consiste di un singolo carattere fra virgolette seguito dai due punti (:).

```
10: "A": PRINT "PRIMO"
20: END
80: "B": PRINT "SECONDO"
90: END
```

Per etichettare un programma si possono usare i seguenti caratteri: A, S, D, F, G, H, J, K, L, ,, Z, X, C, V, B, N, M, ed il tasto SPC (spazio). Queste lettere corrispondono ai tasti compresi nelle ultime due righe della tastiera la cui area per facilitarVi, è più scura rispetto al resto della tastiera.

* Per eseguire questi programmi al posto di scrivere RUN 80 o GOTO 10 (in modo RUN) premete il tasto [DEF] ed il tasto corrispondente alla lettera utilizzata nell'etichetta. In questo esempio premendo [DEF] e B apparirà la parola "SECONDO".

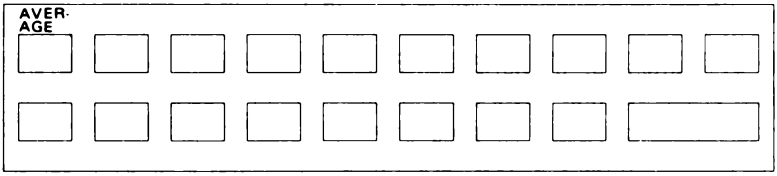
Quando si usa il tasto DEF per iniziare un programma, variabili ed altri controlli non vengono azzerati.
La stessa cosa succede utilizzando il comando GOTO (vedere capitolo 8).

Mascherina

Con il **computer** vi viene data una mascherina, questa mascherina vi aiuterà a ricordare i valori assegnati ai tasti frequentemente usati con il DEF.

Per esempio, se avete un gruppo di programmi che usate spesso contemporaneamente, etichettate i programmi con le lettere e segnate la mascherina in modo da poter facilmente cominciare l'esecuzione di uno qualsiasi dei programmi premendo i due tasti.

Esempio:



CAPITOLO 7
USO DELL'INTERFACCIA STAMPANTE/ REGISTRATORE A CASSETTE CE-126P

L'interfaccia stampante/registratore a cassette **CE-126P** vi consente di aggiungere una stampante e di collegare un registratore a cassette al **computer**. Caratteristiche della **CE-126P:**

* Stampante termica a 24 caratteri per riga.
* Comodo dispositivo di alimentazione della carta e barra di strappo.
* Stampa simultanea dei calcoli se lo si desidera. (Eccetto in modo CAL)
* Facile controllo del formato di visualizzazione o di stampa in BASIC.
* Interfaccia per registratore incorporata, con funzione remote.
* Controllo manuale e da programma del registratore per memorizzare i programmi e i dati.
* Funzionamento con batterie a secco per rendere autonoma l'unità.

Per collegare il **computer** alla **CE-126P**, consultate il manuale di istruzioni fornito insieme alla **CE-126P**.

Uso della stampante

Se usate il **computer** per il calcolo manuale, potete usare la **CE-126P** per stampare simultaneamente i calcoli.

ATTENZIONE:
Il risultato ottenuto mediante la funzione di calcolo diretto nel calcolo manuale non può essere stampato. Non è possibile stampare neppure il calcolo nel modo CAL.

La stampa simultanea viene facilmente ottenuta premendo il tasto [SHIFT] e quindi il tasto [ENTER] (P ⟷ NP) nel modo RUN.
L'indicatore della stampante (un trattino) compare sopra a "PRINT" nell'area inferiore di destra del visore. Dopo di ciò, quando premete [ENTER] alla fine di un calcolo, il contenuto del visore verrà stampato su di una riga ed i risultati verranno stampati sulla riga successiva. Per esempio:

Inserite	Carta
300 / 50 [ENTER]	300/50 6.

Potete ottenere la stampa di risultati o commenti di un programma in Basic, utilizzando l'istruzione LPRINT (vedere capitolo 8). LPRINT da gli stessi risultati dell'istruzione PRINT. La sola differenza si ha quando si ha la visualizzazione di un numero maggiore di 16 caratteri. In questo caso il display mostrerà solo 16 caratteri mentre se si ultilizza la stampa, i caratteri in eccesso verranno automaticamente stampati su di una seconda e, al limite terza riga.

I programmi realizzati con il solo comando PRINT, possono essere convertiti ed usati con la stampante includendo nel programma l'istruzione PRINT = LPRINT (vedere capitolo 8 per i dettagli). Tutti i PRINT che seguiranno si comporteranno come dei LPRINT. L'istruzione PRINT = PRINT annulla questa condizione. Questa istruzione la si può inserire in un programma dove un IF chiede la scelta tra visualizzazione e stampa.

Potete, inoltre, ottenere il listato dei Vostri programmi per mezzo del comando LLIST (vedere capitolo 8).
Se usato senza specificare il numero di riga, il comando LLIST provoca la stampa di tutto il contenuto della memoria. La stampa inizierà con la riga avente il più basso numero d'identificazione e proseguirà in progressione.
Si può ottenere la stampa di un gruppo di linee e se una riga di programma contiene più di 24 caratteri, verrà scritta su più righe.
La seconda riga e quelle successive verranno rientrate di 4 o 6 caratteri, in mdo che il numero della riga risulti più distintamente (riga da 1 a 999: quattro; oltre il 999: sei).

PRECAUZIONI

- Nel caso di un errore a causa di inceppamento della carta (codice di errore 8), tagliare la carta ed estrarre quella nel meccanismo di stampa tirandola. Premete il tasto [C·CE] per cancellare la situazione d'errore.

- Se l'interfaccia stampante/registratore è sottoposta a forti disturbi elettrici esterni, potrebbe stampare numeri a caso. Se questo si dovesse verificare, premete il tasto [BRK] per interrompere la stampa quindi premete il tasto [C·CE] . Il tasto [C·CE] riporta la stampante nelle sue normali condizioni.

  (Quando, durante la stampa, si ha un avanzamento irregolare della carta o si ha un forte disturbo elettrico esterno si può avere un blocco della stampante e la parola "BUSY" sul computer rimane accesa. In questo caso premete il tasto [BRK] per arrestare la stampa. (Eliminate l'inceppamento della carta.) Quindi spegnete e riaccendete la **CE-126P** e premete il tasto [C·CE] .)

- Quando la **CE-126P** non è in uso, spegnete l'interruttore della stampante per preservare le batterie.

Uso dell'interfaccia registratore a cassette

L'uso di questa interfaccia vi consente di memorizzare i programmi ed i dati dal computer su di una cassetta standard (naturalmente vi serve anche un registratore a cassette quale quello da noi venduto per questo computer tascabile: il registratore a cassette opzionale modello **CE-152**). Una volta che i programmi ed i dati sono stati memorizzati su nastro, potete ricaricarli nel computer con una procedura molto semplice.

Collegamento della CE-126P ad un registratore a cassette

Sono necessarie soltanto tre connessioni:

1. Inserite la spina rossa nella presa MIC per il microfono sul registratore a cassette.
2. Inserite la spina grigia nella presa EAR della cuffia sul registratore a cassette.
3. Inserite la spina nera nella presa REM (distanza) sul registratore a cassette.

Registratore a cassette

Vi consigliamo di usare il registratore a cassette opzionale **CE-152** per il vostro computer tascabile. Il **CE-152**, studiato per essere collegato al **computer**, registra i programmi ed i dati tramite l'interfaccia **CE-126P**. Tutti i programmi registrati possono essere reperiti e ricaricati nel **computer**.

Nel caso usiate un registratore a cassette diverso dal **CE-152**, diamo qui di seguito una descrizione delle specifiche minime necessarie per l'interfaccia con la **CE-126P**:

Elemento	Requisiti
1. Tipo di registratore	Qualsiasi registratore, a cassetta, micro-cassetta, o a bobine può essere usato in accordo con i requisiti espressi sotto.
2. Presa d'ingresso	Il registratore dovrebbe avere una mini-presa d'ingresso etichettata "MIC". Non usate mai la presa "AUX".
3. Impedenza d'ingresso	La presa d'ingresso dovrebbe avere una bassa impedenza d'ingresso. (200 ~ 1000 Ohm.)
4. Livello minimo d'ingresso	Minore 3mV o −50 dB.
5. Presa d'uscita	Dovrebbe essere una mini-presa con etichetta "EXT.", "MONITOR", "EAR" o equivalente.
6. Presa d'uscita	Dovrebbe essere inferiore a 10 Ohm.
7. Impedenza d'uscita	Dovrebbe essere superiore a 1V (uscita massima pratica superiore a 100mW)
8. Distorsione	Dovrebbe essere contenuta entro il 15% nella gamma da 2 kHz a 4 kHz.
9. Wow e Flutter	0,3% massimo (W. R. M. S)
10. Altro	La velocità del motore del registratore non deve fluttuare.

* Nel caso che la mini-presa fornita con il **CE-126P** non sia compatibile con le prese ingresso/uscita del vostro registratore, delle prese speciali di conversione sono disponibili in commercio.

NOTA:

- Certi registratori a nastro potrebbero non connettersi bene per via delle loro specificazioni diverse. I registratori soggetti a distorsione sonora, maggior rumore ed a deterioramento dell'alimentazione dovuti a lunghi anni di uso potrebbero non dimostrare risultati soddisfacenti per via dei cambiamenti nelle loro caratteristiche.

Uso dell'interfaccia e del registratore

Registrazione (salvataggio) su nastro magnetico

Vedere le note relative al nastro

1. Portate su OFF l'interruttore REMOTE sulla **CE-126P**.
2. Immettete nel computer un programma o dei dati.
3. Inserite un nastro nel registratore.
   Determinate la posizione sul nastro in cui volete registrare il programma.
   - Quando usate un nastro, accertatevi di fare avanzare il nastro oltre la coda neutra (mylar non magnetico).
   - Quando usate un nastro già parzialmente registrato, cercate una posizione non registrata.
4. Collegate la spina rossa dell'interfaccia alla presa MIC del registratore a cassette e la spina nera alla presa REM.
5. Attivate l'interrutore REMOTE.
6. Premete simultaneamente il tasto RECORD (registrazione) e PLAY (riproduzione) sul registratore a cassette (per metterlo nel modo registrazione).
7. Date le istruzioni di registrazione (istruzione CSAVE, istruzione PRINT #), e premete il tasto [ENTER] per l'esecuzione.

   Impostate dapprima il computer nel modo "RUN" o "PRO". Premete quindi i tasti seguenti:
   [C] [S] [A] [V] [E] [SHIFT] ["] nome del programma [SHIFT] ["] [ENTER] .
   (Per registrare su nastro il contenuto della memoria dei dati, premete i tasti seguenti: [P] [R] [I] [N] [T] [SHIFT] [#] [ENTER] .)
   Es. [C] [S] [A] [V] [E] [SHIFT] ["] [A] [A] [SHIFT] ["] [ENTER]

   Quando premete il tasto [ENTER] , il nastro comincia a svolgersi, ma per circa 8 secondi non viene registrato niente e si sente un segnale acustico. Dopodiché, il nome del programma ed il contenuto della memoria vengono registrati.

8. Quando la registrazione è terminata, viene visualizzato il simbolo di pronto (>) ed il registratore si arresta automaticamente. Ora il vostro programma è su nastro pur permanendo anche nel computer tascabile.
   Quando i dati devono essere registrati automaticamente mediante l'esecuzione del programma (istruzione PRINT, non operazione manuale) fate le operazioni previste dai punti da 1 a 6 prima di eseguire il programma.

   Per aiutarvi a localizzare i programmi sui nastri, servitevi del contanastro del registratore.

Verifica dei programmi

Vedere le note relative ai nastri.

Dopo aver caricato o trasferito un programma al nastro o dal nastro, potete verificare che il programma sul nastro ed il programma nel computer tascabile siano identici (ed accertarvi inoltre che tutto vada bene prima di continuare la programmazione o l'esecuzione dei programmi).

1. Portate su OFF l'interruttore REMOTE.
2. Con la cassetta nel registratore, ultilizzando i tasti di controllo del movimento del nastro portate il nastro nel punto immediatamente precedente il nome del programma da controllare.
3. Inserite la spina grigia nella presa EAR (cuffia) e la spina nera nella presa REM.
4. Portate su ON l'interruttore REMOTE.
5. Premete il tasto PLAY del registratore.
6. Scrivete l'istruzione CLOAD? e cominciate l'esecuzione con il tasto [ENTER]
   Fate come segue:
   Impostate l'unità su "RUN".
   Battete la seguente sequenza di tasti:

   Nome del programma usato precedentemente

   [C] [L] [O] [A] [D] [SHIFT] [?] [SHIFT] ["] [A] [A] [SHIFT] ["] [ENTER]

   Il computer tascabile cerca automaticamente il nome del programma specificato e confronta il contenuto su nastro con il contenuto in memoria.

   Durante il controllo appare un "*" sull'estrema destra del visore. Il simbolo "*" sparisce quando il controllo è completato. Fino a quando il programma non è stato trovato, non appare il simbolo "*" in quanto il controllo non è iniziato.

   (La stessa cosa succede leggendo il primo programma senza un nome di file.)
   Se la macchina verifica che i programmi sono identici, viene visualizzato il simbolo (>) sul visore del computer tascabile.

   Se i programmi sono diversi, l'esecuzione viene interrotta e viene visualizzato ERROR 8. Se ciò si verifica, ripetete la procedura dal punto 1.

Caricamento da nastro magnetico

Vedere le note sui nastri.

Per caricare, trasferire o leggere i programmi ed i dati dal nastro nel computer, usate la procedura seguente.

1. Mettete su OFF l'interruttore REMOTE.

2. Inserite il nastro nel registratore. Posizionate il nastro sul punto immediatamente precedente la parte da leggere.
3. Inserite la spina grigia nella presa EAR del registratore, e la spina nera nella presa REM.
   (Se usate un registratore a cassette che non ha la presa REM, premete il tasto PAUSE per ottenere un arresto temporaneo).
4. Mettete su ON l'interruttore REMOTE.
5. Premete il tasto PLAY sul registratore (per mettere l'unità nel modo di riproduzione)

   Impostate il controllo del VOLUME a medio o massimo.

   Se il registratore non funziona correttamente con il volume regolato al massimo, ridurre il volume e riprovare.

   Impostate il tono sul massimo.

6. Date le istruzioni di trasferimento (istruzione CLOAD, istruzione INPUT #), e premete il tasto [ENTER] per l'esecuzione.

   Mettete il computer nel modo "RUN", premete quindi i tasti seguenti: [C] [L] [O] [A] [D] [SHIFT] ["] nome del programma [SHIFT] ["] [ENTER] . (Per caricare il contenuto della memoria, premete i tasti seguenti: [I] [N] [P] [U] [T] [SHIFT] [#] [ENTER] .)

   Per esempio: modo "RUN"
   [C] [L] [O] [A] [D] [SHIFT] ["] [A] [A] [SHIFT] ["] [ENTER]

   Il nome del programma specificato verrà cercato automaticamente ed il suo contenuto verrà trasferito nel computer.
   Il simbolo "*" apparirà quando il programma richiesto sarà in fase di trasferimento dal nastro al computer.
   (Lo stesso succede se si comanda una lettura senza specificare il nome del programma)
   Il simbolo "*" sparisce quando il programma è stato caricato.

7. Quando il programma è stato trasferito, il computer arresta automaticamente lo svolgimen to del nastro e visualizza il simbolo di pronto (>).

   Per trasferire i dati (istruzione INPUT #) durante l'esecuzione di un programma, fate le operazioni previste nei punti da 1 a 5 prima di eseguire il programma.

   Note:
   - Se si verifica un errore (viene visualizzato ERROR 8), ricominciate la procedura dall'inizio. Se l'errore persiste, alzate od abbassate leggermente il volume.

- Se il codice di errore non viene visualizzato ma lo svolgimento del nastro prosegue (mentre il computer visualizza il simbolo "BUSY"), il trasferimento non è corretto. Premete il tasto ON/BRK per arrestare il nastro. Ripetete la procedura.

- Se l'errore persiste o il nastro continua a svolgersi dopo diversi tentativi di risolvere il problema, pulite e smagnetizzate la testina del registratore.

Note sul nastro

1) Per qualsiasi trasferimento o collazione (verifica del contenuto del nastro rispetto al contenuto della memoria del computer), usate il registratore a cassette utilizzato per la registrazione. Se il registratore per il trasferimento o la collazione è diverso da quello usato per la registrazione, le operazioni di trasferimento o di collazione non sono possibili.

2) Usate sempre soltanto nastri di prima qualità per la memorizzazione dei programmi e dei dati (i nastri di tipo economico possono non avere le caratteristiche necessarie per la registrazione digitale).

3) Tenete pulite le testine di registrazione e di lettura e le parti del registratore — usate la cassetta apposita per la pulizia della testina.

4) Impostazione del volume — impostate il volume al livello medio o massimo.
   Il livello del volume può essere molto importante quando i dati vengono letti dal registratore; alzate od abbassate leggermente il volume in modo da ottenere un trasferimento dei dati senza errori. Talvolta basta alzare o diminuire leggermente il volume per ottenere dei risultati perfetti.

5) Accertatevi che tutti i collegamenti tra il computer e l'interfaccia siano saldi. Accertatevi che i collegamenti tra l'interfaccia ed il registratore siano saldi.

6) Se si verificano dei problemi usando la corrente alternata per il registratore, sostituite la corrente di rete con le batterie (talvolta la corrente di rete aggiunge un certo "ronzio" al segnale che impedisce una corretta registrazione dei dati).
   - Prima di collegare l'adattatore di corrente alla **CE-126P**, spegnerla e quindi eseguire il collegamento.

7) Controllo del tono — impostate il livello massimo.

8) Quando registrate dei programmi o dei dati su di un nastro usato, cancellate la parte prima di effettuare la registrazione dando il comando relativo. (Accertatavi che il programma precedente sia completamente cancellato e che non sia rimasta nessuna parte).

CAPITOLO 8
IL BASIC

Il seguente capitolo è diviso in 3 parti:

COMANDI: Istruzioni eseguite al di fuori di un programma per controllarlo, per modificare certe funzioni, o per utilizzare particolari caratteristiche.

VERBI: Parole d'azione per costruire il programma in BASIC.

FUNZIONI: Operatori speciali usati nel BASIC per modificare una variabile in un'altra.

Comandi e verbi sono in ordine alfabetico e ad ognuno è assegnata una pagina per una più facile consultazione.

Il contenuto di ogni parte è indicato nella tabella sottoriportata in maniera che possiate facilmente identificare a quale categoria appartiene un determinato operatore.

Le funzioni sono raggruppate in 3 categorie e suddivise alfabeticamente per categoria.

Comandi

Controllo del programma

CONT
DELETE
GOTO*
NEW
RENUM
RUN

Controllo del registratore

CLOAD
CLOAD?
CSAVE
INPUT #*
MERGE
PRINT #*

Debugging

LIST
LLIST
TROFF*
TRON*

Controllo delle variabili

CLEAR*
MEM*
DIM*

Controllo delle misure degli angoli

DEGREE*
GRAD*
RADIAN*

Altri

BEEP*
MDF*
PASS
RANDOM*
USING*
WAIT*

* I comandi con questo asterisco sono anche verbi. Le loro funzioni sono identiche sia utilizzandoli come comandi che come verbi. Pertanto vengono descritti nella sezione dedicata ai verbi.

Verbi

Controllo

CHAIN
END
FOR...TO...STEP
GOSUB
GOTO
IF...THEN
NEXT
ON...GOSUB
ON...GOTO
RETURN
STOP

Assegnazione e dichiarazione

CLEAR
DIM
LET

Input e Output

AREAD
CSAVE
DATA
INPUT
INPUT #
LPRINT
PAUSE
PRINT
PRINT #
READ
RESTORE
USING
WAIT

Altri

BEEP
DEGREE
GRAD
MDF
RADIAN
RANDOM
REM
TROFF
TRON

Funzioni

Pseudovariabili

INKEY$
MEM
PI

Funzioni stringa

ASC
CHR$
LEFT$
LEN
MID$
RIGHT$
STR$
VAL

Funzioni numeriche

ABS
ACS
AHC
AHS
AHT
ASN
ATN
COS
CUR
DEG
DMS
EXP
FACT
HCS
HSN
HTN
INT
LN
LOG
POL
RCP
REC
RND
ROT
SGN
SIN
SQR
SQU
TAN
TEN

COMANDI

1 **CLOAD**
2 **CLOAD** "nome del file"

Abbreviazioni: CLO., CLOA.

Vedere: CLOAD?, CSAVE, MERGE, PASS

Scopi

Il comando CLOAD (cassette-LOAD = carica la cassetta) viene utilizzato per trasferire un programma registrato su cassetta.

Uso

1) CLOAD = in questa forma il comando cancella i programmi esistenti dalla memoria e carica il primo programma memorizzato su nastro.

2) CLOAD = "nome del file" (nome del programma). In questa seconda forma dapprima cancella tutta la memoria, quindi cerca sul nastro il programma che ha per nome quello inserito e una volta trovatolo, lo carica in memoria.

Se il **computer** è nel modo PRO o RUN, i programmi in memoria vengono caricati dal nastro.

Esempio

CLOAD Carica il primo programma che trova sul nastro.
CLOAD "PRO3" Ricerca e carica il programma che ha per nome "PRO3".

Note:

1. Se il computer non trova il programma richiesto con il comando CLOAD "nome del file", continuerà a cercare il nome del file anche quando il nastro termina. In questo caso, arrestate la funzione di ricerca premendo il tasto ON BRK. Questo vale anche per i comandi CLOAD?, INPUT #, MERGE, e CHAIN che verranno descritti in seguito.

2. Se dovesse verificarsi un errore durante l'esecuzione del comando CLOAD o CHAIN (descritti in seguito), il programma nel computer non sarà corretto.

3. Durante il caricamento nella posizione di estrema destra del visore viene visualizzato il segno " * ". Il segno " * " scompare al termine del caricamento. Durante la ricerca del programma avente il nome indicato, il segno " * " non viene visualizzato poiché il caricamento non è ancora iniziato.

   (Lo stesso accade quando si legge il primo programma senza alcun nome).

1 **CLOAD?**
2 **CLOAD?** "nome del file"

Abbreviazioni: CLO.?, CLOA.?

Vedere: CLOAD, CSAVE, MERGE, PASS

Scopi

Il comando CLOAD? viene utilizzato per confrontare un programma registrato su nastro con quello contenuto nella memoria del computer.

Uso

Per verificare che un programma sia stato memorizzato correttamente, riavvolgete il nastro fino all'inizio del programma e utilizzate il comando CLOAD?.

1) Il comando CLOAD? confronta il contenuto della memoria con il primo programma che incontra sul nastro, partendo dalla posizione corrente.

2) Nella forma CLOAD? "nome del file" ricerca sul nastro il programma che comincia con il nome indicato e, una volta trovatolo, lo confronta con il programma contenuto in memoria.

Esempio

CLOAD?	Confronta il primo programma che trova sul nastro con quello in memoria.
CLOAD? "PRO3"	Ricerca il programma che ha per nome "PRO3" e lo confronta con quello in memoria.

* Mentre il programma viene verificato, nell'estremità destra del display appare un asterisco. Esso sparisce al termine della verifica, e viene sostituito dal prompt.

1 **CONT**

Abbreviazioni: C., CO., CON.

Vedere anche: RUN, STOP

Scopi

Il comando CONT (continua) viene utilizzato per far continuare un programma che è stato temporaneamente interrotto.

Uso

Quando in un programma viene incontrato il comando STOP, per continuare scrivere CONT.

Lo stesso discorso vale anche quando un programma viene fermato premendo il tasto [BRK] ; per continuarlo, scrivere CONT.
CONT funziona anche quando il programma è temporaneamente interrotto a causa di un comando come PRINT.

Esempio

CONT continua un programma precedentemente interrotto.

1 **CSAVE**
2 **CSAVE** "nome del file"
3 **CSAVE**, "parola chiave"
4 **CSAVE** "nome del file", "parola chiave"

Abbreviazioni: CS., CSA., CSAV.

Vedere: CLOAD, CLOAD?, MERGE, PASS

Scopi

Il comando CSAVE (cassette save = salva su cassetta) viene utilizzato per trasferire (salvare) un programma dalla memoria del computer ad una cassetta magnetica.

Uso

CSAVE comanda il trasferimento su nastro di tutti i programmi contenuti nella memoria senza uno specifico nome.

CSAVE "nome del file" comanda il trasferimento su nastro di tutti i programmi nella memoria assegnando come nome quello indicato tra virgolette.

CSAVE "parola chiave" – questa terza forma comanda il trasferimento di tutti i programmi dalla memoria al nastro senza assegnare un nome al programma ma assegnandogli una parola chiave. Chiunque può in seguito ricaricare il programma, ma non potrà né listarlo, né modificarlo a meno che non conosca la "parola chiave" (vedere descrizione del comando PASS).

CSAVE "nome del file" "parola chiave" comanda il trasferimento della memoria al nastro di tutti i programmi contenuti, assegnando un nome ed una parola chiave (Vedi sopra).

Esempio

CSAVE "PRO3", "SEGRETO" Salva i programmi attualmente in memoria su nastro con il nome "PRO3" e li protegge con la parola chiave "SEGRETO".

1 **DELETE** [numero della prima riga][, [numero della ultima riga]]

Abbreviazioni: DEL., DELE., DELET.

Vedere anche: NEW, PASS

Scopi

Il comando DELETE è utilizzato per cancellare una o pi" righe di un programma.

Uso

Se vengono specificate sia la prima che l'ultima riga da cancellare, ogni riga compresa fra le due date viene cancellata. Se una delle due righe specificate non viene trovate, si ha un errore.
Se viene specificata la sola linea di inizio, solo questa viene cancellata.

Se vengono specificate la riga di inizio e la virgola (,), la linea di inizio e tutte quelle successive vengono cancellate.

Se vengono specificate la virgola (,) e la riga di fine, tutte le linee dall'inizio al numero di fine vengono cancellate.

Se vengono omessi sia il numero della riga di inizio che quello della riga di fine, si ha un errore.

Se il comando DELETE viene eseguito mentre un file di programmi viene letto col comando MERGE, il comando DELETE agisce sul programma letto per ultimo.

Se una "parola chiave" fa parte del programma, il comando DELETE viene ignorato.

Esempio

DELETE 1ØØ Cancella la linea 100 e tutte quelle successive.

1 **GOTO** espressione

Abbreviazioni: G., GO., GOT.

Vedere: RUN

Scopi

Questo comando viene utilizzato per ordinare l'inizio di un programma.

Uso

Per far cominciare un programma al numero di riga specificato dall'espressione, si può utilizzare il comando GOTO (GOTO = va a .....) al posto del comando RUN.

GOTO differisce da RUN da cinque punti di vista:

1) Non viene azzerato il valore dell'intervallo WAIT (vedere WAIT)
2) Il formato stabilito con l'istruzione USING non viene azzerato (vedere USING)
3) I valori contenuti nelle variabili e nelle matrici di variabili vengono mantenuti.
4) L'uguaglianza dei comandi PRINT = LPRINT non è annullata
5) Il puntatore per READ non è stato azzerato.

L'esecuzione di un programma con il comando GOTO è identica all'esecuzione con il tasto [DEF] .

Esempio

GOTO 100 Viene visualizzata la riga 100.

1 **LIST**
2 **LIST** "numero di riga"
3 **LIST** "etichetta"

Abbreviazioni: L., LI., LIS.

Vedere anche: LLIST

Scopi

Il comando LIST (LIST = LISTA) è utilizzato per visualizzare un programma.

Uso

Il comando LIST funziona solo in PRO.

* List visualizza il programma partendo dalla prima riga fino a riempimento del visore.

* List "numero di riga" visualizza il programma dalla riga corrispondente al numero specificato fino a riempimento del visore.

  Se la riga specificata non esiste, il programma sarà visualizzato a partire dalla riga esistente corrispondente al numero maggiore successivo a quello specificato.

* List "etichetta" visualizza il programma partendo dalla riga che porta l'etichetta specificata fino a riempimento del visore.

* Quando più programmi vengono concatenati con il comando MERGE, il comando LIST agisce sull'ultimo programma.

  Se però l'etichetta specificata in LIST etichetta, non esiste nell'ultimo programma, questa viene ricercata in sequenza a partire dal primo programma. Se l'etichetta specificata viene trovata, viene visualizzata la riga che la contiene. Se è stata impostata una parola chiave il comando LIST viene ignorato.

Esempio

LIST 100 Viene visualizzata la riga 100.

1 **LLIST**
2 **LLIST** espressione
3 **LLIST** espressione 1, espressione 2
4 **LLIST** espressione,
5 **LLIST**, espressione

Abbreviazioni: LL., LLI., LLIS.

Vedere: LIST

Scopi

Il comando LLIST viene usato per stampare un programma sulla stampante facoltativa.

Uso

Il comando LLIST può essere usato nel modo PRO o nel modo RUN.

LLIST stampa tutti i programmi in memoria.

LLIST espressione stampa soltanto la riga di programma il cui numero viene dato dall'espressione.

LLIST espressione 1, espressione 2, comanda la stampa del programma a partire dalla riga avente come numero di identificazione quello corrispondente all'espressione 1 o quello più vicino per eccesso e termina alla riga avente il numero di identificazione corrispondente all'espressione 2 o a quello più vicino per eccesso. Vi devono sempre essere almeno 2 righe tra i due numeri.

LLIST espressione, comanda la stampa di tutte le righe di programma a partire dalla riga il cui numero è dato dall'espressione.

LLIST, espressione comanda la stampa di tutte le righe di programma sino alla riga il cui numero è dato dall'espressione, inclusa quest'ultima.

* Quando più programmi vengono fusi usando il comando MERGE il comando LLIST agisce sull'ultimo programma. Per stampare un programma memorizzato prima, eseguite

LLIST "etichetta".

Se è stata impostata una parola chiave, il comando LLIST viene ignorato.

Esempio

LLIST 100, 200 Stampa le righe a partire dal 100 sino alla 200.

1 **MERGE**
2 **MERGE** "nome del file"
(agisce per le operazioni manuali in modo RUN e PRO)

Abbreviazioni: MER., MERG.

Vedere: CLOAD

Scopi

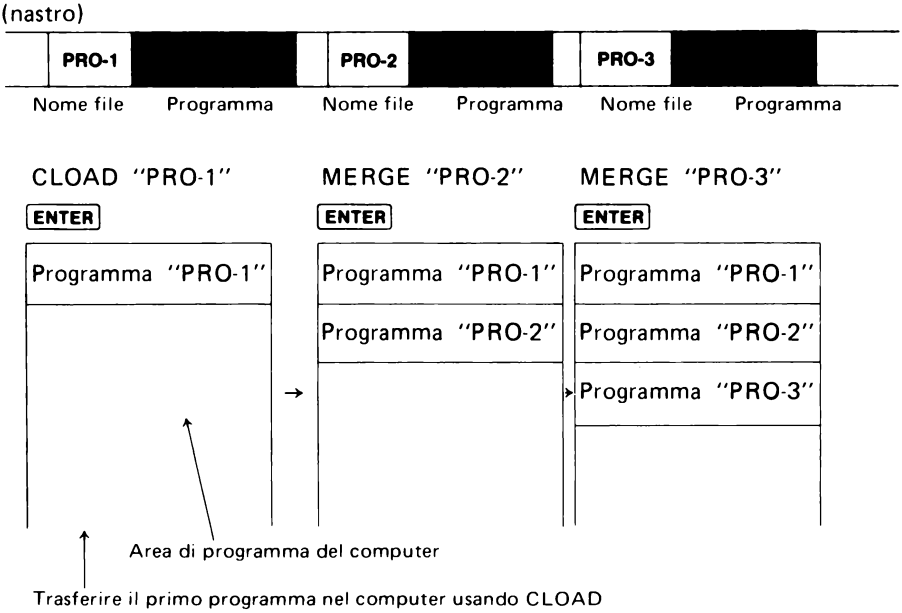
Il comando MERGE viene utilizzato per trasferire in memoria un programma memorizzato su nastro magnetico e unirlo a un programma già presente in memoria.

Uso

Il comando MERGE conserva il programma già memorizzato nel **computer** e inoltre trasferisce in memoria un programma memorizzato sul nastro. Perciò molti programmi diversi possono essere memorizzati contemporaneamente nel **computer**.

Esempio

Quando dovete memorizzare i programmi PRO-1 a PRO-3, PRO-1 viene memorizzato usando il comando CLOAD, mentre PRO-2 e PRO-3 vengono trasferiti nel computer utilizzando il comando MERGE. Lo stato della memoria è il seguente:



I programmi caricati con il comando MERGE vengono memorizzati come nell'esempio. I programmi sono trattati come descritto qui di seguito utilizzando i loro numeri di riga.

- Se il primo numero di riga del programma caricato con il comando MERGE è maggiore dell'ultimo numero di riga del programma caricato precedentemente, i due programmi vengono considerati un solo programma.
- Se il primo numero di riga del programma caricato con il comando MERGE è minore dell'ultimo numero di riga del programma precedentemente caricato, i due programmi vengono considerati separati.

  Nell'esempio precedente, in cui i numeri di riga dei programmi PRO-1, PRO-2 e PRO-3 sono rispettivamente 10-200, 50-150 e 160-300, PRO-1 e PRO-2 vengono considerati separatamente. PRO-2 e PRO-3 vengono considerati un unico programma con i numeri di riga 50-300.

* Il caricamento dei programmi con il comando MERGE può dare origine a due o più programmi con gli stessi numeri di riga. In questo caso l'esecuzione di RUN o GOTO (RUN "espressione", GOTO "espressione") è valida solo per l'ultimo programma. Non è assolutamente possibile eseguire i programmi precedenti. Aggiungete pertanto un'etichetta all'inizio del programma da eseguire e eseguitelo utilizzando il tasto definito.

  Notate comunque che solo l'ultimo programma di quelli unificati può essere modificato dopo che è stato eseguito il comando MERGE e che i programmi caricati in precedenza non possono essere modificati. Perciò, aggiungete un'etichetta al programma prima di unirlo al programma successivo.

Funzione di programmi protetti da parole chiave

Quando caricate dei programmi con parole chiave (programmi protetti da parole chiave) utilizzando il comando MERGE, il trattamento dei programmi è diverso a seconda che siano protetti o meno all'interno del computer.

Programmi protetti

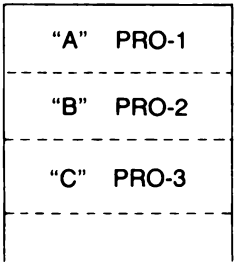
I programmi protetti da parola chiave non possono essere caricati.

Programmi non protetti

Se vengono caricati con il comando MERGE dei programmi protetti da parola chiave tutti i programmi esistenti nel computer diventano protetti.

Quando i programmi esistenti nel computer sono protetti, anche i programmi senza parola chiave diventano protetti da parola chiave se caricati col comando MERGE.

Esecuzione di programmi fusi



La figura illustra la memoria quando effettuate il caricamento di PRO-1 seguito dal caricamento di PRO-2 e PRO-3 con il comando MERGE. Se avviate un programma utilizzando RUN o GOTO (RUN "espressione" o GOTO "espressione"), verrà eseguito PRO-3. D'altro canto, se avviate il programma utilizzando RUN "etichetta", GOTO "etichetta", o un tasto definito, il sistema cerca l'etichetta specificata dall'inizio di PRO-3 all'interno del computer. Se questa non viene trovata in PRO-3 la ricerca continua in PRO-1.

Se non viene trovata neppure in PRO-1 viene ricercata in PRO-2. Se l'etichetta viene trovata, il programma viene eseguito dalla riga etichettata.

Nota che poiché l'etichetta viene ricercata in questo modo, se un'etichetta utilizzata in PRO-1 e in PRO-2 viene utilizzata anche in PRO-3, PRO-1 e PRO-2 non possono essere eseguiti.

1 **NEW**

Abbreviazione: nessuna

Vedere: CLEAR, PASS

Scopi

Il comando NEW (nuovo) viene utilizzato per cancellare il programma esistente.

Uso

Quando viene utilizzato nel modo PRO, il comando NEW cancella tutti programmi ed i dati attualmente in memoria (I programmi protetti da parola chiave non possono essere cancellati).

Il comando NEW non è funzionante nel modo RUN e, se viene utilizzato, causa ERROR 9.

Esempio

NEW cancella il programma o i dati.

1 **PASS** "stringa di caratteri"

Abbreviaz.: PA., PAS.

Vedere: CLOAD, CSAVE, DELETE NEW, RENUM

Scopi

Il comando PASS (PASSARE = OLTREPASSARE) è utilizzato per inserire o cancellare una parola chiave.

Uso

Le parole chiave vengono utilizzate per proteggere i programmi dal controllo o dalle modifiche di altri utenti.
Una parola chiave consiste in una stringa non può lunga di 7 caratteri, scelti tra le lettere dell'alfabeto e tra i seguenti simboli: ! # $ % & ( ) * + – / , . : ; < = > ? @ √ π ^

Nota: Non usate alcuna istruzione o verbo BASIC come parola d'ordine.

Una volta che è stato dato il comando PASS, tutti i programmi in memoria sono protetti. Quando un programma è protetto, non è possibile esaminarlo o modificarlo e non è possibile listarlo o visualizzarlo coi comandi LIST o LLIST, o aggiungere o togliere righe intere.

Se vi sono differenti programmi in memoria, inserendo il comando PASS, tutti risulteranno protetti.

Per togliere la protezione è sufficiente inserire nuovamente il comando PASS con l'identica parola chiave.

Nota: Quando viene dichiarata una parola chiave con 7 o più caratteri, solo i primi 7 caratteri sono validi e vengono usati per impostare o togliere la protezione.
Premete [ENTER] dopo la parola chiave.
Se scrivete dei caratteri o dei simboli dopo la parola chiave otterrete un messaggio di errore e la parola chiave non può essere usata.
(esempio) PASS"ABCDEFG":A=123 [ENTER] → **ERROR 1**

Esempio

PASS "SEGRETO" — Stabilisce quale parola chiave, a protezione di tutti i programmi in memoria, la parola SEGRETO (i programmi sono protetti).

1 **RENUM** [nuovo numero di riga][,[vecchio numero di riga] [,incremento]]

Abbreviazioni: REN., RENUM.,

Scopi

Il comando RENUM viene usato per dare un nuovo numero a righe di programma. Questo comando è efficace nell'uso manuale nel modo PRO (Program).

Uso

Questo comando cambia i numeri delle linee di programma a partire dalla linea specificata cambiandoli della quantità specificata.

Se vengono omessi i dati sul nuovo numero di linea e sull'incremento, viene assunto per ambedue il valore 10. Se il vecchio numero non viene indicato, la renumerazione parte dalla prima linea del programma. Se il vecchio numero di linea specificato non viene trovato, si ha un errore.
Esempio 1 : RENUM
Renumera tutte le linee di programma di 10 unità a partire dalla riga 10.
Esempio 2 : RENUM 100, 50, 10
Cambia il vecchio numero di linea 50 nel nuovo numero di linea 100 e renumera le linee successive di 10 unità.

Il comando RENUM cambia automaticamente tutti i riferimenti ai numeri di riga seguenti GOTO, GOSUB, IF ~ THEN, ON ~ GOTO, ON ~ GOSUB, RESTORE ecc. in accordo con i nuovi valori. In questo caso, tuttavia, si avrà un errore se l'espressione (ad es. GOTO 2*50). Se a causa di una scorretta designazione della riga si dovesse verificare un errore, la renumerazione della riga il cui numero è scorretto non può essere effettuata dal comando RENUM. In tale caso, riscrivete temporaneamente il comando contenente il numero di riga scorretto cambiandolo in un'istruzione REM e poi correggetelo (cambiandolo per esempio con ON ~ GOTO) dopo l'esecuzione del comando RENUM.

Il comando RENUM non può venire eseguito se le righe da renumerare superano le 65279 o se la specificazione richiede una variazione dell'ordine di esecuzione delle righe del programma (ad esempio, se si tenta di eseguire RENUM 15, 30 quando le tre righe di programma 10, 20 e 30 già esistono.

L'esecuzione del comando RENUM in un programma vasto richiede un certo tempo. Se premete il tasto [BRK] per interrompere il programma mentre sulla destra del visore è presente un asterisco (*), il programma ritornerà alla condizione in cui si trovava prima dell'esecuzione di RENUM. Tuttavia, una simile interruzione da parte di [BRK] viene ignorata se sul visore appaiono due asterischi (**).

L'area di lavoro del "numero di righe di programma × 4 bytes" viene usata solo quando il comando RENUM viene eseguito. Renumerando le righe di programma, cambiano i numeri di riferimento di riga di GOTO, GOSUB ecc. Come rislutata, la nuova versione del programmapotrebbe richiedere un numero superiore di byte. In altre parole, la nuova riga GOTO 200 usa un byte in pi" rispetto alla vecchia riga GOTO 20. Il comando RENUM non è eseguibile se l'area di lavoro residua è insufficiente a causa dell'eccessivo numero di byte usati. In tali casi, cancellate dalla memoria delle variabili col comando CLEAR e potreste allora essere in grado di eseguire il comando.

(Vedere l'appendice A per quanto riguarda i messaggi di errore relativi a RENUM)

1 **RUN**
2 **RUN** numero di riga

Abbreviazioni: R., RU.

Vedere: GOTO, MERGE

Scopi

Il comando RUN (letteralmente: CORRI) ordina al **computer** l'esecuzione del programma in memoria.

Uso

Il comando RUN ordina l'esecuzione del programma in memoria partendo dalla riga avente il più piccolo numero di identificazione.

Il comando RUN (numero di riga) comanda l'esecuzione di un programma che comincia con il numero di riga specificato.

* Quando più programmi vengono fusi col comando MERGE, l'ultimo programma concatenato sarà eseguito con RUN (1) o RUN "numero di riga".

RUN differisce dal comando GOTO per le seguenti cinque caratteristiche:

1) Il valore dell'intervallo WAIT viene azzerato (vedere WAIT)
2) Il formato stabilito con USING viene cancellato (vedere USING).
3) Le variabili e le matrici di variabili, oltre che delle 26 variabili fisse (A–Z) vengono azzerate.
4) Viene impostata la condizione di uguaglianza PRINT = PRINT.
5) Il contatore di READ viene azzerato all'inizio dell'istruzione DATA.

L'esecuzione di un programma con il comando GOTO è uguale all'esecuzione con il tasto DEF. In tutte e tre le forme di esecuzione del programma l'intercalamento FOR/NEXT e GOSUB è annullato.

Esempio

RUN 100 Esegue il programma a partire dalla riga 100.

VERBI

1 **AREAD** nome della variabile

Abbreviaz.: A., AR., ARE., AREA.

Vedere: INPUT e USO DEL TASTO DEF AL CAP. 6

Scopi

Il verbo AREAD (AUTO READ = AUTO LETTURA) è utilizzato per far inserire in una variabile quanto presente sul visore; il programma contenente il verbo AREAD deve cominciare utilizzando il tasto [DEF] (iniziando con il RUN o il GOTO si cancellerebbe il visore rendendo così impossibile l'operazione di lettura).

Uso

Quando il programma è etichettato e pertanto se ne comanda l'esecuzione per mezzo del tasto [DEF] , si può utilizzare il verbo AREAD per assegnare ad una variabile un valore o dei caratteri, presenti sul visore, senza dover ricorrere al verbo input.

Il verbo AREAD deve essere collocato nella prima riga, immediatamente dopo l'etichetta. Se inserito in altre parti del programma verrà ignorato. In un programma si può utilizzare solo una volta il comando AREAD.

Per utilizzare il verbo AREAD, col computer in RUN, scrivete il valore desiderato e premete il tasto [DEF] seguito dalla lettera che identifica il programma.

Se si deve inserire una stringa, non è necessario porla tra virgolette.

Esempio

```
10 "X": AREAD N
20 PRINT N^2
30 END
```

Inserendo 7 e premendo i tasti [DEF] e X avrete la visualizzazione del risultato (49).

Note:

1. Quando il visore visualizza il simbolo (">") all'inizio dell'esecuzione del programma, la variabile indicata viene annullata.

2. Quando il contenuto viene visualizzato dal verbo PRINT all'inizio dell'esecuzione del programma, viene memorizzato quanto segue:

Esempio: Quando il seguente programma è eseguito;

```
10 "A": PRINT "ABC", "DEFG"
20 "S": AREAD A$: PRINT A$
modo RUN
[DEF] [A] → ABC        DEFG
[DEF] [S] → DEFG
```

- Quando il risultato visualizzato dal primo programma ha come risultato due numeri (PRINT espressione numerica, espressione numerica) o due stringhe (PRINT stringa, stringa), verrà memorizzato il risultato di destra.
- Quando il risultato visualizzato dal primo programma ha vari numeri (PRINT espressione numerica, espressione numerica, espressione numerica, espressione numerica . . .), verrà memorizzato il primo di sinistra.
- Quando il risultato visualizzato dal primo programma ha diverse stringhe (PRINT stringa, stringa, stringa, . . .) la stringa che compare per ultima viene memorizzata.

1 **BEEP** espressione

Abbreviaz.: B., BE., BEE.,

Scopi

Il verbo BEEP (BEEP = Parola onomatopeica) comanda l'emissione di una nota udibile.

Uso

Il verbo BEEP comanda l'emissione di una o più note a 4 kHz.

Il numero delle note è determinato dall'espressione che segue il verbo BEEP e che deve essere numerica (un numero positivo minore di 9.999999999E+99). Inoltre per determinare il numero di ripetizioni del suono, viene considerata solo la parte intera.

Il verbo BEEP può essere utilizzato anche come comando, in questo caso il suono inizierà non appena premuto il tasto [ENTER] .

Esempio

```
10 A=5 : B$="9"
20 BEEP 3          (Produce 3 BEEP)
30 BEEP A          (Produce 5 BEEP)
40 BEEP (A+4)/2    (Produce 4 BEEP)
50 BEEP B$         Questo non è corretto e causa ERROR 9
60 BEEP -4         Questo non è corretto ma non causa errori.
```

1 **CHAIN**
2 **CHAIN** espressione
3 **CHAIN** "nome del file"
4 **CHAIN** "nome del file, espressione"

Abbreviaz.: CHA., CHAI.

Vedere: CLOAD, CSAVE, e RUN

Scopi

Il verbo CHAIN viene usato per iniziare l'esecuzione di un programma che è stato memorizzato su nastro. Può essere utilizzato soltanto in combinazione con l'interfaccia opzionale CE-126P e CE-152.

Uso

Il verbo CHAIN viene utilizzato quando devono essere memorizzati su cassetta uno o più programmi. Durante l'esecuzione di un programma, quando compare il verbo CHAIN, un programma viene caricato dalla cassetta e eseguito.

CHAIN carica il primo programma memorizzato su nastro e inizia l'esecuzione dal numero di riga più basso del programma. In questo caso CHAIN ha lo stesso effetto di CLOAD e RUN nel modo RUN.

CHAIN espressione carica il primo programma memorizzato su nastro e inizia l'esecuzione dal numero di riga specificato nell'espressione.

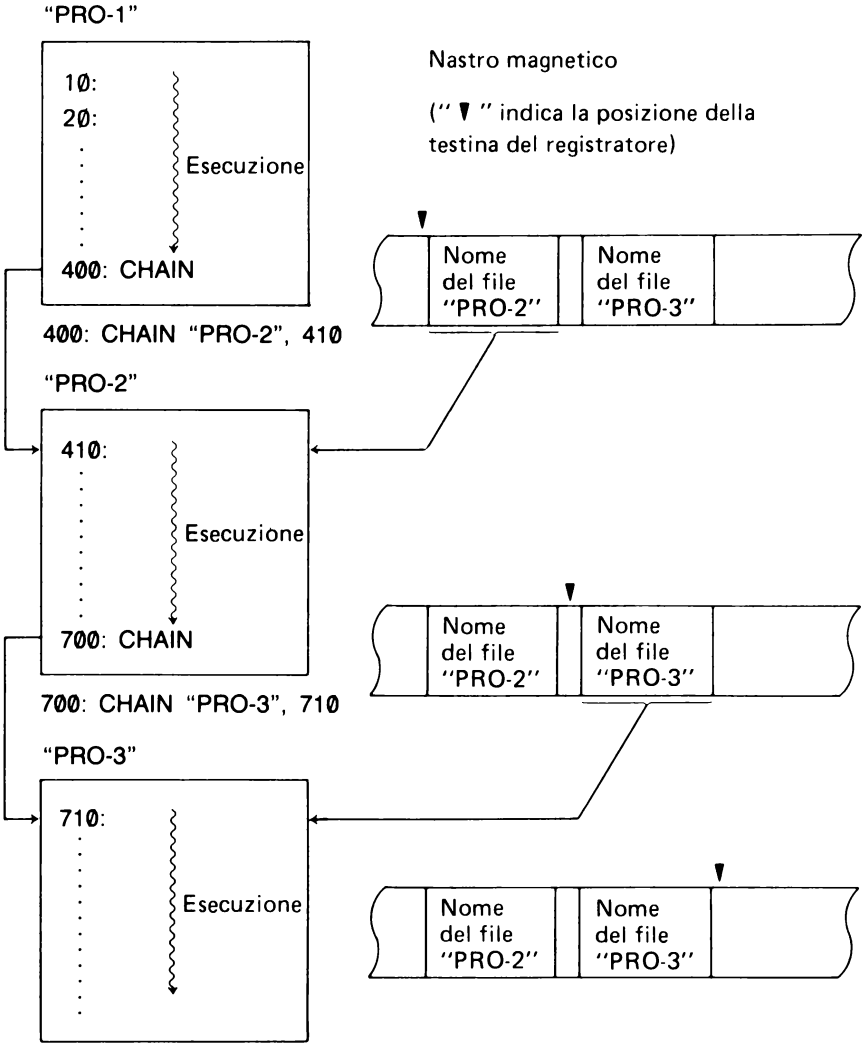
CHAIN "nome del file" ricerca sul nastro il programma il cui nome è indicato da "nome del file", carica tale programma, e inizia l'esecuzione dal numero di riga inferiore.

CHAIN "nome del file", espressione ricerca sul nastro il programma il cui nome è indicato dal nome del file, carica il programma e si inizierà l'esecuzione dal numero di riga indicato dall'espressione.

Esempio

1Ø CHAIN	Carica il primo programma da nastro e inizia l'esecuzione dal numero di riga più basso.
2Ø CHAIN "PRO-2", 48Ø	Ricerca sul nastro il programma chiamato PRO-2, lo carica, e ne inizia l'esecuzione dal numero di riga 48Ø.

Supponiamo per esempio di avere tre parti di programma chiamate PRO-1, PRO-2, PRO-3. Ciascuna di queste parti termina con un'istruzione CHAIN.



990: END

Durante l'esecuzione, quando il computer incontra l'istruzione CHAIN, la parte successiva viene richiamata in memoria e eseguita. In questo modo tutte le parti vengono infine eseguite.

Nota: Quando un programma contenente un comando CHAIN viene caricato da nastro tramite il comando MERGE, controllate che il comando CHAIN sia corretto.

1 **CLEAR**

Abbreviazioni: CL., CLE., CLEA.

Vedere: DIM

Scopi

Il verbo CLEAR (liberare) viene usato per cancellare tutte le variabili usate nel programma e per porre tutte le variabili preassegnate a zero o NUL.

Uso

Il verbo CLEAR consente di ricuperare spazio utilizzato per la memorizzazione di variabili, il che potrebbe essere opportuno per esempio quando le variabili usate nella prima parte di un programma non sono necessarie nella seconda parte e lo spazio disponibile è limitato. CLEAR può essere usato anche all'inizio di un programma quando parecchi programmi sono residenti in memoria e si vuole liberare lo spazio usato per l'esecuzione dei programmi precedenti.

CLEAR non libera lo spazio usato dalle variabili A–Z, A$–Z$, o A(1)–A(26) (senza dichiarazione DIM) poiché queste sono assegnate permanentemente (vedere capitolo 4). CLEAR pone a zero le variabili numeriche e pone a NUL le variabili stringhe.

Esempio

```
10 A=5: DIM C(5)
20 CLEAR          Libera lo spazio assegnato a C(  ) e pone A a zero.
```

1 **DATA** elenco di espressioni
Dove: Elenco di espressioni è: espressione
o: espressione, elenco di espressioni

Abbreviaz.: DA., DAT.,

Vedere: READ, RESTORE

Scopi

Il verbo DATA (DATI) è da utilizzarsi in unione al verbo READ e fornisce a quest ultimo i valori necessari.

Uso

Quando dovete assegnare dei valori iniziali ad una matrice, è opportuno elencare i valori in un'istruzione DATA ed usare un'istruzione READ in un ciclo FOR ... NEXT per caricare tali valori nella matrice. Quando viene eseguito il primo READ, viene letto il primo valore che compare nella prima istruzione DATA. I READ successivi usano i valori successivi nell'ordine progressivo in cui essi compaiono nel programma, indipendentemente da quanti valori sono elencati in ciascuna istruzione DATA o da quante istruzioni DATA vengono usate.

Le istruzioni DATA non hanno alcun effetto se vengono incontrate nel corso della normale esecuzione di un programma, pertanto esse possono essere inserite dovunque sembri opportuno. Molti programmatori amano includerle immediatamente dopo l'istruzione READ che le usa. Se lo si desidera, i valori contenuti nell'istruzione DATA possono essere letti una seconda volta utilizzando l'istruzione RESTORE.

Esempio

```
10 DIM B(10)                Prepara una matrice.
20 WAIT 128
30 FOR I=1 TO 10
40 READ B(I)                Carica i valori dall'istruzione DATA in B( )
                              per cui B(1) sarà 10, B(2) 20, B(3) 30 ecc.
50 PRINT B(I)               Visualizza i dati della matrice B(I).
50 NEXT I
70 DATA 10,20,30,40,50,60
80 DATA 70,80,90,100
90 END
```

1 **DEGREE**

Abbreviazioni: DE., DEG., DEGR., DEGRE.

Vedere: GRAD e RADIAN

Scopi

Il verbo DEGREE (grado) viene usato per convertire il valore di un angolo in gradi decimali.

Uso

Il **computer** utilizza tre forme diverse per la rappresentazione delle unità di misura degli angoli: gradi sessagesimali, radianti, gradienti. Queste forme vengono usate per specificare l'argomento delle funzioni SIN, COS e TAN e per dare i risultati ottenuti dalle funzioni ASN, ACS ed ATN.

La funzione DEGREE cambia la forma di rappresentazione di tutti i valori angolari in gradi decimali sino a che non viene usato un verbo GRAD o RADIAN. Le funzioni DMS e DEG possono essere usate per convertire i gradi decimali nella forma gradi, minuti, secondi e viceversa.

Esempio

```
10 DEGREE
20 X=ASN 1
30 PRINT X
```

X ha ora il valore di 90, cioè 90 gradi, l'arcoseno di 1.

1 **DIM** elenco dimensioni

In cui: elenco dimensioni	è: specifica dimensioni
	o: spec. dimens., elenco dimens.
e: specifica dimensioni	è: spec. dimens. numeriche
	o: spec. dimens. di stringa
e: specif. dimens. numeriche	è: valore numerico (dimensione)
e: specif. dimens. di stringa	è: nome di stringa (dimensioni)
	o: nome di stringa (dim.) * lungh.
e: valore numerico	è: nome di variab. numer. valido
e: nome di stringa	è: nome di stringa variab. valido
e: dimensioni	è: dimensione
	o: dimensione, dimensione
e: dimensione	è: numero di elementi
e: lunghezza	è: lunghezza di ogni stringa in una matrice di stringhe

Abbreviazioni: D., DI.

Scopi

Il verbo DIM (dimension = dimensione) viene usato per riservare lo spazio per le matrici di variabili numeriche o di stringa.

Uso

Il verbo DIM deve essere usato per riservare lo spazio per delle matrici di variabili, a meno che siano della forma A( ) o A$( ), e per le variabili semplici come A1 e B2$.

Il numero massimo di dimensioni consentito in una matrice è 2; l'ampiezza massima di una dimensione è 255. Oltre al numero di elementi specificati nell'istruzione di dimensione, viene riservato un ulteriore elemento corrispondente alla posizione Ø. Per esempio, con DIM B(3) riserviamo B(Ø), B(1), B(2), B(3). Nelle matrici bidimensionali esiste un'ulteriore colonna ed un'ulteriore fila "zero".

Nelle matrici di stringhe, si specifica la dimensione di ogni elemento di stringa oltre al numero degli elementi. Per esempio, DIM B$(3) * 12 riserva lo spazio per 4 stringhe ciascuna delle quali ha una lunghezza massima di 12 caratteri. Se non viene specificata la lunghezza, ogni stringa può contenere un massimo di 16 caratteri.

Quando viene dimensionata una matrice numerica, tutti i valori vengono inizialmente impostati a zero; in una matrice di stringhe i valori sono impostati a NULLO.

Per la dichiarazione DIM delle matrici A ed A$, fate riferimento al paragrafo relativo alle variabili.

Le matrici di variabili possono essere azzerate (in modo, cioè, che non siano più definite) per mezzo del comando CLEAR. Dando inizio ad un programma con RUN, tutte variabili vengono automaticamente azzerate.
Un nome di variabile usato una volta non può essere usato una seconda volta. Eseguendo di nuovo un programma usando il comando GOTO, dopo aver usato il tasto [DEF], il medesimo nome di variabile dichiarato precedentemente viene dichiarato di nuovo, se viene eseguita la riga con il comando DIM. In questo caso, azzerate la matrice di variabili per mezzo del comando CLEAR e dichiaratela di nuovo.

Esempio

1Ø DIM B(1Ø)	Riserva lo spazio per una matrice numerica di 11 elementi.
2Ø DIM C$(4, 4) * 1Ø	Riserva lo spazio per una matrice di stringhe bidimensionale composta da 5 righe e 5 colonne; ogni stringa avrà una lunghezza massima di 1Ø caratteri.

1 **END**

Abbreviazioni: E., EN.

Scopi

Il verbo END (FINE) è utilizzato per indicare la fine di un programma.

Uso

Quando nella memoria del **computer** sono inclusi vari programmi, per evitare che il computer, dopo aver elaborato un programma, continui con il successivo, è opportuno indicare il punto di fine d'ogni programma. Questo lo si ottiene inserendo, dopo l'ultima istruzione, il verbo END.

Esempio

```
10 PRINT "CIAO"
20 END
30 PRINT "ARRIVEDERCI"
40 END
```

Con questo programma in memoria, comandando il RUN 10, si otterrà la visualizzazione della parola CIAO mentre non verrà visualizzata la parola ARRIVEDERCI. Con RUN 30 otterremo la visualizzazione della parola ARRIVEDERCI.

1 **FOR** variabile numerica = espressione 1 **TO** espressione 2
2 **FOR** variabile numerica = espressione 1 **TO** espressione 2
STEP espressione 3

Abbreviaz.: F. e FO.; STE.

Vedere: NEXT

Scopi

Il verbo FOR (PER) TO (A) è utilizzato in unione al verbo NEXT (SUCCESSIVO) per ripetere una serie di operazioni un numero determinato di volte.

Uso

Il verbo FOR ed il verbo NEXT sono utilizzati in coppia per includere un gruppo d'istruzioni che devono essere ripetute. La prima volta che questo gruppo di istruzioni viene eseguito la variabile di ciclo (la variabile che compare immediatamente dopo la parola FOR) ha il valore dell'espressione 1.

Ogni volta che l'esecuzione raggiunge il verbo NEXT la variabile di ciclo viene incrementata del valore dello STEP (passo) e quindi questo valore viene confrontato con l'espressione 2. Se il valore della variabile di ciclo è minore o uguale all'espressione 2, le istruzioni comprese nel ciclo di vengono nuovamente eseguite, partendo dall'istruzione successiva al FOR. In FOR variabile numerica = espressione 1 TO espressione 2 lo step è 1; in FOR variabile numerica = espressione 1 TO espressione 2 STEP lo step è dato dall'espressione 3. Se il valore della variabile di ciclo è maggiore dell'espressione 2, l'esecuzione continua con l'istruzione immediatamente successiva a NEXT. Poiché il confronto viene fatto alla fine, le istruzioni all'interno di una coppia FOR-NEXT vengono sempre eseguite almeno una volta. L'espressione 1, l'espressione 2 e l'espressione 3 devono essere nella gamma compresa tra −9.999999999E99 e 9.999999999E99. Quando l'espressione 3 è zero, l'anello (ciclo) FOR/NEXT sarà infinito.

La variabile di ciclo può essere utilizzata nell'ambito del gruppo di istruzioni, per esempio come indice di una matrice, ma è necessario fare attenzione quando cambiate il valore della variabile di ciclo.

Non dovete mai scrivere dei programmi che saltino dall'esterno di una coppia FOR/NEXT ad un'istruzione all'interno di una coppia FOR/NEXT. Analogamente i programmi non devono mai uscire da un ciclo FOR/NEXT se non tramite l'istruzione NEXT. Per ottenere ciò impostate la variabile di ciclo a un valore maggiore dell'espressione 2.

Il gruppo di istruzioni racchiuso da una coppia FOR/NEXT può comprendere un'altra coppia di istruzioni FOR/NEXT utilizzanti una variabile di ciclo diversa purché tale coppia sia completamente racchiusa, cioé se il gruppo comprende un'istruzione FOR, deve comprendere un'istruzione NEXT corrispondente. Le coppie FOR/NEXT possono essere "annidate" fino ad un massimo di cinque livelli.

Esempio

```
10 FOR I=1 TO 5
20 PRINT I
30 NEXT I
```

Questo gruppo di istruzioni stampa i numeri 1, 2, 3, 4, 5.

```
40 FOR N=10 TO 0 STEP-1
50 PRINT N
60 NEXT N
```

Questo gruppo di istruzioni effettua un conteggio alla rovescia 10, 9, 8, 7, 6, 5, 4, 3, 2 ,1, 0.

```
70 FOR N=1 TO 10
80 X=1
90 FOR F=1 TO N
100 X=X*F
110 NEXT F
120 PRINT X
130 NEXT N
```

Questo gruppo di istruzioni calcola e stampa N fattoriale per i numeri da 1 a 10.

Nota: L'esecuzione del ciclo FOR-NEXT non termina anche se si esce dal ciclo. Notate quindi che potete incorrere in un errore di annidamento del il ciclo FOR-NEXT (ERROR 5) a seconda del tipo di programma (programmi che eseguono il comando FOR un certo numero di volte).

1 **GOSUB** espressione

Abbreviazioni: GOS., GOSU.

Vedere: GOTO, ON...GOSUB, ON...GOTO, RETURN

Scopi

Il verbo GOSUB (va alla subroutine) viene usato per eseguire una subroutine BASIC.

Uso

Quando desiderate eseguire lo stesso gruppo di istruzioni diverse volte nel corso di un programma o desiderate utilizzare una serie di istruzioni scritte in precedenza in diversi programmi, è opportuno usare una subroutine BASIC avvalendosi dei verbi GOSUB e RETURN.

Il gruppo di istruzioni che compongono la subroutine è incluso nel programma in una posizione in cui non vengono raggiunte nella sequenza normale di esecuzione. Una posizione frequentemente usata è quella che segue l'istruzione END che segna la fine del programma principale. In tali posizioni del corpo principale del programma, (dove devono essere eseguite le subroutine), includete un'istruzione GOSUB con un'espressione che indica il numero della riga iniziale della subroutine. L'ultima riga della subroutine deve essere un'istruzione RETURN. Quando viene eseguita l'istruzione GOSUB, il **computer** passa al numero di riga indicato ed elabora le istruzioni sino a che arriva ad un'istruzione RETURN. A questo punto il controllo ritorna all'istruzione che segue l'istruzione GOSUB.

Le subroutine possono includere un GOSUB. Le subroutine possono essere inserite una dentro l'altra fino a 10 livelli.

L'espressione in una istruzione GOSUB non può includere una virgola, per esempio non si può usare "A(1, 2)". Poiché esiste una struttura ON...GOSUB per scegliere diverse subroutine in determinate posizioni del programma, l'espressione è generalmente costituita soltanto dal numero di riga desiderato. Quando usate un'espressione numerica, questa deve dare un numero di riga valido, cioè da 1 a 65279, diversamente incorrerete nell'ERROR 4.

Esempio

```
10 GOSUB 100
20 END
100 PRINT "CIAO"
110 RETURN
```

Facendo girare questo programma, verrà visualizzata una volta la parola "CIAO".

1 **GOTO** espressione

Abbreviazioni: G., GO., GOT.

Vedere: GOSUB, ON...GOSUB, ON...GOTO

Scopi

Il verbo GOTO trasferisce l'esecuzione del programma ad un numero di riga specificato.

Uso

Il verbo GOTO fa saltare il programma da una posizione all'altra. A differenza dal verbo GOSUB, il verbo GOTO non ricorda la posizione dalla quale ha avuto luogo il trasferimento.

L'espressione in un'istruzione GOTO non può includere una virgola, per esempio non è possibile usare "A(1, 2)". Poiché esiste una struttura ON...GOTO per scegliere diverse destinazioni in determinate posizioni del programma, l'espressione generalmente è costituita soltanto dal numero di riga desiderato. Quando usate un'espressione numerica, questa deve indicare un numero di riga valido, cioè 1–65279, diversamente incorrerete in un ERROR 4.

I programmi ben progettati generalmente vanno semplicemente dall'inizio alla fine, con l'eccezione delle subroutine eseguite durante il programma. Pertanto, l'uso principale del verbo GOTO trova applicazione come parte di un'istruzione IF...THEN.

Esempio

```
10 INPUT A$
20 IF A$="Y" THEN GOTO 50
30 PRINT "NO"
40 GOTO 60
50 PRINT "SI"
60 END
```

Questo programma stampa "SI", se è stata immessa una "Y" e stampa "NO" in qualsiasi altro caso.

1 GRAD

Abbreviaz.: GR., GRA.

Vedere: DEGREE e RADIAN

Scopi

Il verbo GRAD è utilizzato per convertire il valore di un angolo in gradienti.

Uso

Il **computer** ha tre forme per rappresentare i valori angolari (gradi decimali, i radianti ed i gradienti). Queste forme sono utilizzate per specificare l'argomento delle funzioni SIN, COS, TAN, ASN, ACS, e TAN.

La funzione GRAD converte in gradienti tutte le forme angolari fino a che non verrà utilizzato il verbo DEGREE o RADIAN.
Il gradiente rappresenta la misura angolare in termini di percentuale: es. un angolo di 45° è gradiente al 50^g.

Esempio

Per trovare i valori di un seno nelle rispettive unità angolari, specificate prima "D" per i gradi, "R" per i radianti e "G" per i gradi decimali e quindi inserite l'angolo del seno.

```
10 INPUT "DEG=D,RAD=R,GRAD=G?";A$
20 IF A$="D" THEN 100
30 IF A$="R" THEN 200
40 GRAD :GOSUB 300:GOTO 40
100 DEGREE :GOSUB 300:GOTO 100
200 RADIAN :GOSUB 300:GOTO 200
300 INPUT "SIN ?";B
310 PRINT "SIN" ;B;"=";SIN B
320 RETURN
```

1 **IF** condizione **THEN** istruzione
2 **IF** condizione istruzione

Abbreviazioni: nessuna per IF, T., TH., THE.

Scopi

La coppia di verbi IF (SE) ... THEN (in questo caso) viene utilizzato per fare eseguire o non fare eseguire un'istruzione a seconda delle condizioni al momento in cui viene eseguito il programma.

Uso

Nell'esecuzione normale di un programma in BASIC, le istruzioni vengono eseguite nella sequenza in cui si presentano. La coppia di verbi IF...THEN consente di prendere delle decisioni durante l'esecuzione in modo che una data istruzione venga eseguita soltanto quando lo si desidera. Quando la condizione che fa parte dell'istruzione IF è vera, l'istruzione viene eseguita; quando è falsa, l'istruzione viene saltata.

La condizione contenuta nell'istruzione IF può essere una qualsiasi delle espressioni di relazione viste nel capitolo 4, oppure un'espressione numerica sebbene l'intenzione dell'istruzione sia in tal caso meno chiara. Qualsiasi espressione che ha come risultato uno zero o un numero negativo, è considerata falsa; tutte le espressioni che danno come risultato un numero positivo vengono considerate vere.

L'istruzione che segue il THEN, può essere qualsiasi istruzione in BASIC, incluso un altro IF...THEN. Se è un'istruzione LET, deve comparire il verbo LET.

Le due forme dell'istruzione IF hanno un'azione identica, ma la prima è più chiara.

Esempio

```
10 INPUT "CONTINUO?"; A$
20 IF A$="SI" THEN GOTO 10
30 IF A$="NO" THEN GOTO 60
40 PRINT "RISPONDERE SI O NO"
50 GOTO 10
60 END
```

Questo programma continuerà a chiedere "CONTINUO?" fino a quando risponderete "SI"; si arresterà se risponderete "NO", se darete una risposta diversa da "SI" o "NO", visualizzerà "RISPONDERE SI O NO".

1 **INPUT** lista di variabili

Dove:	Lista d'inserimento	è: gruppo d'inserimento
		o: gruppo d'inserimento, lista d'inserimento
e:	gruppo d'inserimento	è: lista di variabili
		o: prompt, lista di variabili
		o: prompt; lista di variabili
e:	lista di variabili	è: variabile
		o: variabile, lista di variabili
e:	prompt	è: qualsiasi costante stringa

Abbreviazioni: I., IN., INP., INPU.

Vedere: INPUT #, READ, PRINT

Scopi

Il verbo INPUT è utilizzato per assegnare dalla tastiera dei valori a delle variabili.

Uso

Quando desiderate assegnare dei valori diversi a delle variabili per ottenere successive elaborazioni si usa il verbo INPUT.

Nella sua forma più semplice, il verbo INPUT è immediatamente seguito dal nome di una variabile numerica o di stringa (INPUT A o INPUT A$). In questo caso quando il programma incontra questa istruzione, farà apparire sulla sinistra del visore, un punto interrogativo (attendo l'inserimento); scritto il dato e premuto il tasto [ENTER] viene effettuata l'assegnazione del dato ed il programma continua o con la richiesta di un nuovo inserimento, nel caso il verbo INPUT preveda vari inserimenti o con l'istruzione successiva.

Se nell'istruzione INPUT è incluso un commento (prompt), il processo è esattamente lo stesso, con l'eccezione che, invece del punto interrogativo, alla sinistra del visore viene visualizzata la stringa di commento. Se la stringa di commento è seguita da un punto e virgola, il cursore viene visualizzato immediatamente dopo il commento. Se il commento è seguito da una virgola, esso viene visualizzato, quindi, quando premete un tasto, il visore viene azzerato e alla sinistra viene visualizzato il primo carattere del dato immesso.

Quando specificate un commento e la lista seguente contiene più di una variabile, la seconda variabile e quelle successive vengono segnalate con il punto interrogativo. Se nella lista è inclusa un secondo commento, questo viene visualizzato prima della variabile immediatamente successiva.

Se premete il tasto [ENTER] senza aver immesso alcun dato, la variabile mantiene il valore che aveva prima dell'istruzione INPUT.

Esempio

1Ø INPUT A	Azzera il visore e visualizza un punto interrogativo sulla sinistra del visore.
2Ø INPUT "A="; A	Visualizza "A=" ed attende l'inserimento dei dati.
3Ø INPUT "A=", A	Visualizza "A=". Quando vengono inseriti i dati, "A=" scompare ed i dati vengono visualizzati partendo da sinistra.
4Ø INPUT "X=?"; X, "Y=?"; Y	Visualizza "X=?" ed attende il primo inserimento di dati. Dopo che è stato premuto [ENTER], il visore viene azzerato e alla sinistra viene visualizzato "Y=?".

Nota:
Per cancellare un errore durante l'inserimento di dati per un comando INPUT premete il tasto [C·CE] e poi inserite i dati corretti.

1 **INPUT #** lista di variabili
2 **INPUT #** "nome del file", lista di variabili

Dove: lista di variabili è: variabile
o: variabile, lista di variabili

Abbreviazioni: I. #, IN. #, INP. #, INPU. #

Vedere: INPUT, PRINT #, READ

Scopi

Il verbo INPUT # è utilizzato per caricare, da nastro, uno o più valori.

Uso

* Nome della variabile

Nell'istruzione INPUT # è possibile specificare i seguenti tipi di variabili:
(1) variabili fisse – A, B, C, A(7), D*, A(2Ø)*, ecc.
(2) variabili semplici – AA, B3, CP$, ecc.
(3) matrici di variabili – S(*), HP(*), K$(*), ecc.

1) Trasferimento dei dati alle variabili fisse

Per trasferire i dati dal nastro alle variabili fisse, specificate i nomi delle variabili nelle istruzioni INPUT #.

INPUT # "DATA 1"; A, B, X, Y

Questa istruzione trasferisce i dati dal file su cassetta avente il nome "DATA 1" alle variabili A, B, X ed Y nell'ordine indicato.

Per assegnare a tutte le variabili fisse disponibili e, nel caso vengano definite, alle variabili fisse estese (A(27) ed oltre) i dati trasferiti dal nastro, specificate la prima variabile apponendo un asterisco (*).

INPUT # "D-2"; D*

Questa istruzione trasferisce il contenuto del file su cassetta "D-2" alle variabili D–Z nonché alle variabili A(27) ed oltre.

INPUT # A(1Ø) * (senza dichiarazione DIM)

Questa istruzione trasferisce i dati del primo file trovato dopo che è stato fatto partire il nastro, alle variabili A(1Ø) ed oltre (J–Z ed A(27) ed oltre).

Nota 1. Se una matrice chiamata A è già definita dall'istruzione DIM, non è possibile definire le variabili fisse con suffisso in forma di A( ).

Nota 2. Il trasferimento dei dati alle variabili fisse ed alle variabili fisse estese (A(27) ed oltre) continuerà sino a che si arriva alla fine del file di dati origine sul nastro o sino a che la memoria del computer si satura, provocando un ERROR 6.

2) Trasferimento di dati alle variabili semplici

I dati contenuti nel file sul nastro possono essere trasferiti alle variabili semplici specificando i nomi delle variabili semplici desiderate nell' istruzione INPUT #.

```
INPUT # "DM-1"; AB, Y1, XY$
```

Questa istruzione trasferisce i dati dal file su nastro chiamato "DM-1" alle variabili semplici AB, Y1, ed XY$.

Nota 1. I dati numerici devono essere trasferiti alle variabili semplici numeriche ed i dati alfabetici devono essere trasferiti alle variabili alfabetiche semplici. Non è ammesso il trasferimento incrociato.

Nota 2. Le posizioni delle variabili semplici devono essere riservate nell'area dati del programma prima di eseguire l'istruzione INPUT #. In caso contrario, si verificherà un errore.
Usate le istruzioni di assegnazione per riservare le posizioni per le variabili semplici.

```
AA=Ø [ENTER]
B1$="A" [ENTER]
INPUT # AA, B1$ [ENTER]
```

Usate i valori numerici o i caratteri appropriati nelle istruzioni di assegnazione per riservare le posizioni per le variabili.

3) Trasferimento dei dati alle matrici di variabili

Per trasferire i dati da un archivio su nastro alle matrici di variabili, specificate il nome della matrice nell'istruzione INPUT # sotto forma di nome di matrice (*).

```
5Ø DIM B(5)
6Ø INPUT # "DS-4"; B(*)
```

In questo modo potete trasferire i dati dal file su nastro chiamato "DS-4" alle variabili (B(Ø)—B(5)) nella matrice B.

Nota 1: I dati numerici devono essere trasferiti alle matrici di variabili numeriche aventi la stessa lunghezza di dati; i dati costituiti da caratteri devono essere trasferiti alle matrici variabili costituiti da caratteri aventi la stessa lunghezza di dati. Se questa regola non viene osservata, si incorrerà in un errore.

Nota 2: Le posizioni per le matrici di variabili devono essere riservate nell'area dati del programma prima di eseguire l'istruzione INPUT #. In caso contrario, si avrà un errore. Usate l'istruzione DIM per definire preliminarmente la matrice.

– ATTENZIONE –

Se il numero delle variabili specificate dell'istruzione INPUT # non concorda con la quantità di dati registrati sul nastro, avverrà quanto segue:

* Se il numero di dati registrati nel file su nastro (che devono essere trasferiti) è superiore al numero di variabili specificate, il trasferimento dei dati verrà eseguito sino all'ultima variabile, e i dati restanti verranno ignorati.

* Se il numero di dati registrati nel file su nastro (che devono essere trasferiti) è inferiore al numero delle variabili specificate, tutti i dati del file verranno trasferiti alle variabili sino alla fine del file, e le variabili restanti manterranno il loro contenuto precedente.
  In questo caso tuttavia, il computer continuerà ad aspettare il trasferimento dei dati dal nastro. Per porre termine a questa condizione, dovete premere il tasto ON BRK .

* Se l'istruzione INPUT # viene eseguita senza che sia stato specificato alcun nome di variabile, si incorrerà in un errore (ERROR 1).

1 **LET** variabile = espressione
2 variabile = espressione

Abbreviaz.: LE.

Scopi

Il verbo LET (FAI) è utilizzato per assegnare ad una variabile un valore.

Uso

Il verbo LET assegna il valore di un'espressione alla variabile designata (LET A = 3 + 5) ed ovviamente, il contenuto dell'espressione, deve essere numerico, per assegnazioni a variabili numeriche, e stringa, per variabili stringa.
Nel caso si voglia assegnare ad una variabile numerica una stringa, o viceversa, dovrete far ricorso alle funzioni di conversione STR$ o VAL.
Il verbo LET può essere omesso in tutte le istruzioni d'assegnazione (A = 3 + 5) salvo in quei casi dove il verbo LET è utilizzato dopo l'istruzione THEN del verbo IF...THEN. Solo in questo caso il LET deve essere usato.

Esempio

1Ø I = 1Ø	Assegna ad I il valore 1Ø.
2Ø A = 5 * I	Assegna ad A il valore 5Ø.
3Ø X$ = STR$ (A)	Assegna ad X$ il valore di A convertito da numerico a stringa ('5Ø')
4Ø IF I >= 1Ø THEN LET Y$ = X$ + ".ØØ"	Assegna a Y$ il contenuto della stringa X$ più la stringa ".ØØ" ovvero '5Ø.ØØ'.

Per la stampante CE-126P

1 **LPRINT** { espressione / stringa di caratteri }

2 **LPRINT** { espressione / stringa di caratteri } , { espressione / stringa di caratteri }

3 **LPRINT** { espressione / stringa di caratteri } ; { espressione / stringa di caratteri } ; . . . ; { espressione / stringa di caratteri }

Abbreviazioni: LP., LPR., LPRI., LPRIN.
Vedere: PRINT, USING

Scopi

Il verbo LPRINT (stampare) viene usato per comandare la stampa delle informazione sulla stampante della stampante CE-126P.

Uso

Per inviare nuovamente il comando di stampa del programma sulla stampante, eseguite il comando CLOSE.

Nel formato (1), i numeri vengono allineati a destra, mentre i caratteri alfanumerici vengono stampati verso il lato sinistro della carta. Se una riga contiene più di 24 caratteri, l'avanzamento alla nuova riga viene eseguito automaticamente.

- Nel formato (2), le 24 colonne della riga vengono divise in due gruppi di 12 colonne e i dati vengono stampati in modo simmetrico attorno alla virgola.

Il valore numerico che può essere contenuto in 12 colonne (cifre) viene stampato sull'estrema destra, mentre le stringhe di caratteri vengono stampate a partire dall'estrema sinistra. Se l'espressione da stampare supera le 12 colonne, per i valori numerici vengono stampate le prime 12 cifre troncando le cifre meno significative della frazione decimale, mentre per le stringhe di caratteri vengono stampati i primi 12 caratteri a partire da sinistra.

Nella terza forma, i valori vengono stampati a partire dal margine sinistro del foglio. Se il valore da stampare supera le 24 colonne, la stampa va automaticamente a capo. Si possono effettuare stampe fino ad un massimo di 96 caratteri.

Nota: Non usate alcun comando o verbo BASIC come stringa di espressioni.

Esempio

```
10 A=10: B=20: X$="ABCDE": Y$="XYZ"
20 LPRINT A
30 LPRINT X$
40 LPRINT A, B
50 LPRINT X$; A; B
60 LPRINT
```

1 **MDF** Espressione

Abbreviazione: MD.

Vedere anche: USING

Scopi

Il verbo MDF viene usato per arrotondare il valore di una espressione.

Uso

La funzione MDF viene usata per arrotondare il valore di una espressione al numero di decimali specificato col comando USING.
Questo verbo è efficace solo quando viene specificato un numero di decimali col tasto USING.

Esempio

	Visore
USING "###.###" MDF (Ø.5/9)	**Ø.Ø56**
1Ø USING "###.###" 2Ø A=MDF(5/9) 3Ø PRINT A 4Ø USING 5Ø PRINT A,5/9 6Ø END	
RUN [ENTER]	**Ø.556**
[ENTER]	**Ø.556 5.55555E-Ø1**

1 **NEXT** variabile numerica

Abbreviaz.: N., NE., NEX.

Vedere: FOR

Scopi

Il verbo NEXT viene utilizzato per indicare la fine di un gruppo di istruzioni da ripetere in un ciclo FOR/NEXT.

Uso

L'uso di questo verbo l'abbiamo visto quando abbiamo descritto il verbo FOR. Ricordiamo che la variabile numerica che segue il NEXT deve essere identica alla variabile numerica indicata nel FOR.

Esempio

```
10 FOR I=1 TO 10
20 PRINT I
30 NEXT I
```

Visualizza i numeri da 1 a 10 ogni volta che viene premuto [ENTER] .

1 **ON** espressione **GOSUB** elenco di espressioni

Dove: Elenco di espressioni è: espressione
o: espressione, elenco di espressioni

Abbreviaz.: O.; GOS., GOSU.

Vedere: GOSUB, GOTO, ON...GOTO

Scopi

Il verbo ON...GOSUB permette di elaborare determinate subroutine, in dipendenza del valore dell'espressione che segue l'ON.

Uso

Quando viene eseguito il comando ON...GOSUB, l'espressione compare tra ON e GOSUB viene calcolata e ridotta a numero intero.

Se il valore dell'espressione è 1, viene eseguita la prima subroutine, dell'elenco come in un GOSUB normale. Se il valore è 2, viene eseguita la seconda subroutine dell'elenco e così via. Dopo il RETURN della subroutine, il programma prosegue con l'istruzione che segue il comando ON...GOSUB.

Se il valore dell'espressione è zero, negativo, o più grande del numero di subroutine previsto dall'elenco, nessuna subroutine viene effettuata, ed il programma prosegue con la prima istruzione che segue.

Nota: Non è possibile usare le virgole nelle espressioni che seguono il GOSUB, in quanto il **computer** non è in grado di distinguere le virgole di un'espressione dalle virgole tra le espressioni.

Esempio

```
10 INPUT A
20 ON A GOSUB 100, 200, 300
30 END
100 PRINT "PRIMO"
110 RETURN
200 PRINT "SECONDO"
210 RETURN
300 PRINT "TERZO"
310 RETURN
```

Inserendo 1 si otterrà la visualizzazione di "PRIMO", con 2, "SECONDO" con 3, "TERZO". Ogni altro numero non provocherà alcuna visualizzazione.

1 **ON** espressione **GOTO** elenco di espressioni

Dove: elenco di espressioni è: espressione
o: espressione, elenco di espressioni

Abbreviaz.: O.; G., GO., GOT.

Vedere: GOSUB, GOTO, ON...GOSUB

Scopi

Il verbo ON...GOTO permette di effettuare determinati salti in dipendenza del valore dell'espressione che segue l'ON.

Uso

Tale valore è ridotto al suo intero e solo questa parte viene considerata. Se il valore dell'espressione è 1, verrà effettuata la prima subroutine, la cui posizione nel programma è indicata immediatamente dopo il GOTO, se il valore è 2, verrà effettuata la seconda la cui posizione è indicata di seguito la prima, ecc.

Se l'espressione è zero, negativo, o più grande del numero di salti previsto dall'elenco che segue il GOTO, nessun salto viene effettuato, ed il programma prosegue con la prima istruzione che segue.

Nota: Non è possibile usare le virgole nelle espressioni che seguono il GOTO, in quanto il **computer** non è in grado di distinguere le virgole di un'espressione dalle virgole tra le espressioni.

Esempio

```
10 INPUT A
20 ON A GOTO 100, 200, 300
30 GOTO 900
100 PRINT "PRIMO"
110 GOTO 900
200 PRINT "SECONDO"
210 GOTO 900
300 PRINT "TERZO"
310 GOTO 900
900 END
```

Inserendo 1 si otterrà la visualizzazione di "PRIMO", con 2, "SECONDO", con 3, "TERZO". Ogni altro numero non provocherà alcuna visualizzazione.

1 **PAUSE** espressione
2 **PAUSE** espress., espress.
3 **PAUSE** elenco di stampa; elenco di stampa; . . . ; elenco di stampa

Dove: elenco di stampa è: espressione
o: espressione, elenco di stampa
e: espressione è: espressione
o: clausola USING; espressione

La clausola USING è trattata separatamente, nel capitolo dal titolo omonimo.

Abbreviaz.: PAU., PAUS.

Vedere: LPRINT, PRINT, USING e WAIT

Scopi

Il verbo PAUSE (PAUSA) è utilizzato per visualizzare sul display delle informazioni per un breve periodo.

Uso

Il verbo PAUSE viene usato per visualizzare le informazioni dei messaggi, i risultati dei calcoli, ecc. Il funzionamento di PAUSE è identico a quello di PRINT con l'eccezione che dopo PAUSE, il **computer** attende per un breve intervallo prefissato di circa Ø,85 secondi e quindi continua l'esecuzione del programma senza attendere che venga premuto il tasto [ENTER] o che trascorra l'intervallo WAIT.

La prima forma dell'istruzione PAUSE visualizza un unico valore. Se l'espressione è numerica, il valore viene visualizzato all'estrema destra del visore. Se si tratta di una stringa di caratteri, la visualizzazione comincia dall'estrema sinistra.

Nella seconda il visore è suddiviso in gruppi di 12 colonne. I valori vengono visualizzati in sequenza partendo dal primo valore specificato. Anche in questo caso, in un intervallo di 12colonne, ilvalore numerico di un'espressione viene visualizzato a partire dalla destra del visore, mentre i caratteri vengono visualizzati a partire da sinistra.

- Il numero di valori (elementi) specificato nel formato (2) deve essere compreso tra 2.
- Se il valore specificato supera le 12 colonne avviene quanto segue.
  1) Quando il valore numerico supera le 12 cifre, vengono troncate le cifre meno significative.
  2) Quando i caratteri superano le 12 colonne vengono visualizzati solo i primi 12 caratteri (a partire da sinistra).

Nella terza forma il valore specificato viene visualizzato in modo continuo a partire dal lato sinistro del visore.

Nota: Non usate alcun comando o verbo BASIC come stringa di caratteri in un'istruzione PAUSE.

Esempio

```
10 A = 10; B = 20: X$ = "ABCDEF":
   Y$ = "XYZ"
```

	Visore	
20 PAUSE A		**10.**
30 PAUSE X$	**ABCDEF**	
40 PAUSE X$, B	**ABCDEF**	**20**
50 PAUSE Y$; X$	**XYZABCDEF**	
60 PAUSE A*B		**200.**

1 **PRINT** espressione
2 **PRINT** espress., espress.
3 **PRINT** elenco di stampa
4 **PRINT = LPRINT**
5 **PRINT = PRINT**

Dove: elenco di stampa è: espressione
o: espressione, elenco di stampa
e: espressione è: espressione
o: clausola USING; espressione

La clausola USING è trattata separatamente, nel capitolo dal titolo omonimo.

Abbreviaz.: P., PR., PRI., PRIN.

Vedere: LPRINT, USING e WAIT

Scopi

Il verbo PRINT è utilizzato per visualizzare le informazioni sul visore o stamparle.

Uso

Il verbo PRINT viene utilizzato per visualizzare le informazioni dei messaggi, i risultati dei calcoli, ecc. La prima forma dell'istruzione PRINT visualizza un valore singolo. Se l'espressione è numerica, il valore viene visualizzato all'estrema destra del visore. Se si tratta di un'espressione costituita da una stringa di caratteri, la visualizzazione comincia dall'estrema sinistra.

Nella seconda il visore è suddiviso in gruppi di 12 colonne. I valori vengono visualizzati in sequenza partendo dal primo valore specificato. Anche in questo caso, in un intervallo di 12 colonne, il valore numerico di un'espressione viene visualizzato a partire dalla destra del visore, mentre i caratteri vengono visualizzati a partire da sinistra.

- Il numero di valori specificato nella seconda deve essere compreso nell'intervallo da 2.

- Se il valore specificato supera le 12 colonne avviene quanto segue.
  1) Quando il valore numerico supera le 12 cifre, vengono troncate le cifre meno significative.
  2) Quando i caratteri superano le 12 colonne vengono visualizzati solo i primi 12 caratteri (a partire da sinistra).

Nella terza il valore specificato viene visualizzato in modo continuo a partire dal lato sinistro del visore.

Nota: Non usate alcun comando o verbo BASIC come stringa di caratteri in un'istruzione PRINT.

Esempio

	Visore
1Ø A = 123: B = 456: X$ = "ABCDEF": Y$ = "VWXYZ" 2Ø PRINT X$, B	**ABCDEF 456.**
3Ø PRINT A; B	**123.456.**
4Ø PRINT X$; A	**ABCDEF123.**
5Ø PRINT Y$; B	**VWXYZ456.**

1 **PRINT #** "lista di variabili"
2 **PRINT #** "nome del file"; lista di variabili

Dove: lista di variabili è: variabile
o: variabile, lista di variabili

Abbreviaz.: P.#, PR.#, PRI. #, PRIN. #

Vedere: INPUT#, PRINT, READ

Scopi

Il verbo PRINT # è utilizzato per trasferire su nastro il contenuto delle variabili.

Uso

* Nome di variabile
Per i nomi delle variabili è possible usare i seguenti tipi di variabili:
(1) variabili fisse – A, B, X, A(26), C*, A(1Ø)*, ecc.
(2) variabili semplici – AA, B2, XY$, ecc.
(3) matrici di variabili – B(*), CD(*), N$(*), ecc.

1) Salvataggio del contenuto delle variabili fisse su nastro
Il contenuto delle variabili fisse può essere salvato (trasferito) su nastro specificando i nomi delle variabili desiderate (separati da delle virgole) nell'istruzione PRINT#.

PRINT# "DATA 1"; A, B, X, Y

Questa istruzione trasferisce il contenuto delle variabili A, B, X ed Y sul file su nastro chiamato "DATA 1".

Se desiderate trasferire il contenuto della variabile fissa specificata e di tutte le restanti variabili fisse, aggiungete al nome di tale variabile un asterisco *.

Istruzione PRINT#
L'istruzione PRINT # vi consente di trasferire i dati dalla memoria del computer su nastro.

PRINT# "D-2"; D*	Questa istruzione trasferisce il contenuto delle variabili fisse D–Z, e delle variabili fisse estese A(27) ed oltre, se sono state definite, nel file su nastro chiamato "D-2".
PRINT# E, X$, A(3Ø)*	Questa istruzione trasferisce il contenuto delle variabili fisse E ed X$, nonché delle variabili fisse estese A(3Ø) e di tutte le variabili restanti, sul nastro

Nota: Nell'istruzione PRINT# è possibile specificare i nomi delle variabili fisse con il suffisso A(1)–A(26) nello stesso modo in cui vengono specificati i nomi A–Z (o A$–Z$). Tuttavia, se la matrice A è già stata definita dall'istruzione DIM, non è possibile usare A( ) per definire le variabili fisse con suffisso.

2) Salvataggio del contenuto delle variabili semplici (variabili costituite da due caratteri).

E' possibile trasferire su nastro il contenuto delle variabili semplici specificando i nomi delle variabili desiderate.

PRINT# "DM-1"; AB, Y1, XY$

Questa istruzione trasferisce il contenuto delle variabili semplici AB, Y1, ed XY$ nel file su nastro chiamato "DM-1".

3) Salvataggio del contenuto delle matrici di variabili

Il contenuto di tutte le variabili di una matrice specifica può essere trasferito su nastro specificando il nome della matrice con l'aggiunta di un asterisco racchiuso tra parentesi (*).

PRINT# "DS-2"; X(*), Y$(*)

Questa istruzione trasferisce il contenuto di tutti gli elementi (X(Ø), X(1), ...) della matrice X e di tutti gli elementi (Y$(Ø), Y$(1), ...) della matrice Y$, nel file su nastro chiamato "DS-2".

Nota: Non è possibile trasferire il contenuto soltanto di uno o più elementi specifici di una matrice. Benché le variabili fisse o le variabili fisse con suffisso vi consentano di trasferire soltanto delle loro parti specifiche, una matrice (per esempio A), definita dall'istruzione DIM non vi consente di trasferire soltanto una sua parte specifica.

* Se viene eseguita l'istruzione PRINT# senza specificare alcun nome di variabile, si incorrerà nell'ERROR 1.

ATTENZIONE

Le posizioni per le variabili fisse estese quali A(27) ed oltre, per le variabili semplici e/o per le matrici di variabili devono essere riservate nell'area dati del programma prima di eseguire l'istruzione PRINT#. Diversamente, l'esecuzione dell'istruzione PRINT# per le variabili indefinite porterà ad un errore.

1 **RADIAN**

Abbreviaz.: RAD., RADI., RADIA.

Vedere: DEGREE, GRAD

Scopi

Il verbo RADIAN è utilizzato per convertire i valori angolari in radianti.

Uso

Il **computer** ha tre forme per rappresentare i valori angolari: i gradi decimali, i radianti ed i gradienti.

Queste forme sono utilizzate per specificare l'argomento delle funzioni SIN, COS, TAN, ASN, ACS e ATN.

La funzione RADIAN cambia la forma di tutti i valori angolari precedentemente usati, in radianti. La forma RADIAN rappresenta gli angoli in termini di lunghezza dell'arco rispetto al raggio; es. 360 è 2 PI radianti in quanto la circonferenza di un cerchio e 2 PI volte il raggio.

Esempio

```
10 RADIAN
20 X = ASN 1        X ora ha il valore di 1,570796327 ovvere PI/2, l'arcoseno
                      di 1.
30 PRINT X
```

1 **RANDOM**

Abbreviazioni: RA., RAN., RAND., RANDO.

Scopi

Il verbo RANDOM viene utilizzato per azzerare il numero di partenza per la generazione di numeri casuali.

Uso

Quando vengono generati dei numeri casuali utilizzando la funzione RND, il **computer** ha come base un numero di partenza. La funzione RANDOM azzera questo numero e lo sostituisce con un altro scelto a caso.

Il numero di partenza sarà sempre lo stesso, ogni volta che si accende il **computer**. Di conseguenza, la sequenza della generazione casuale, per mezzo della funzione RND, sarebbe sempre la stessa se non si potesse cambiare il numero di partenza per mezzo della funzione RANDOM. Questo è molto utile quando si esegue l'elaborazione di un programma in quanto significa che il comportamento del programma sarebbe sempre lo stesso ogni volta che viene eseguito anche se include la funzione RND. Quando volete avere dei numeri veramente casuali, è possibile usare l'istruzione RANDOM per ottenere che il numero di partenza stesso sia casuale.

Esempio

```
10 RANDOM
20 X = RND 10
```

Quando il programma viene eseguito dalla riga 20, il valore di X si basa su di un numero di partenza standard. Partendo dalla riga 10, verrà usato un nuovo numero di partenza.

1 **READ** elenco variabili

Dove: elenco di variabili è: variabile
o: variabile, elenco di variabili

Abbreviaz.: REA.

Vedere: DATA, RESTORE

Scopi

Il verbo READ (LEGGI) è utilizzato per leggere i valori indicati dal verbo DATA ed assegnarli a delle variabili.

Uso

Volendo assegnare dei valori iniziali ad una matrice, è opportuno elencare i valori in un'istruzione DATA ed usare un'istruzione READ in un ciclo FOR...NEXT per caricare i valori nella matrice. Quando viene eseguita la prima istruzione READ, viene rimandato il primo valore della prima istruzione DATA. Le successive istruzioni READ usano i valori successivi nell'ordine progressivo in cui compaiono nel programma, indipendentemente da quanti valori sono indicati in ogni istruzione DATA o da quante istruzioni DATA vengono usate.

Se lo si desidera, è possibile leggere una seconda volta i valori contenuti in un'istruzione DATA usando l'istruzione RESTORE.

Esempio

```
10 DIM B (10)          Crea una matrice
20 WAIT 32
30 FOR I=1 TO 10
40 READ B(I)           Carica i valori dall'istruzione DATA nella matrice
50 PRINT B(I)*2;         B( )-B(1) è 10, B(2) è 20, B(3) è 30, ecc.
60 NEXT I
70 DATA 10, 20, 30, 40, 50, 60
80 DATA 70, 80, 90, 100
90 END
```

1 **REM** note

Abbreviaz.: NESSUNA

Scopi

Il verbo REM (REMARK = ANNOTAZIONE) serve per includere dei commenti in un programma.

Uso

Spesso è utile includere in un programma dei commenti esplicativi. Questi commenti possono dare dei titoli, dei nomi di autori, le date delle ultime modifiche, delle note d'uso, delle note sugli logaritmi usati, ecc. Questi commenti vengono inclusi mediante l'istruzione REM.

L'istruzione REM non ha alcuna influenza sull'esecuzione del programma e può essere inclusa in qualsiasi punto dello stesso. Tutto ciò che segue il verbo REM viene interpretato come un commento.

Esempio

```
10 REM   QUESTA RIGA NON INFLUENZA IL PROGRAMMA
```

1 **RESTORE**
2 **RESTORE** espressione

Abbreviaz.: RES., REST., RESTO., RESTOR.

Vedere: DATA, READ

Scopi

Il verbo RESTORE (ripristinare) è utilizzato per rileggere dei dati elencati da un DATA o per cambiare l'ordine col quale questi dati sono letti.

Uso

Di norma, usando il verbo READ, il **computer** comincia la lettura dal primo valore indicato in un'istruzione DATA e procede progressivamente con i valori seguenti.

La prima forma dell'istruzione RESTORE comanda l'azzeramento del contatore posizionandolo sul primo valore della prima istruzione DATA, in modo da poterlo rileggere.

La seconda forma dell'istruzione RESTORE riposiziona il contatore sul primo valore della prima istruzione DATA il cui numero di riga è superiore al valore dell'espressione.

Esempio

```
10 DIM B(10)          Crea una matrice
20 WAIT 32
30 FOR I=1 TO 10
40 RESTORE            Assegna il valore 20 a ciascun elemento di B( ).
50 READ B(I)
60 PRINT B(I)*I
70 NEXT I
80 DATA 20
90 END
```

1 **RETURN**

Abbreviaz.: RE., RET., RETU., RETUR.

Vedere: GOSUB, ON...GOSUB

Scopi

Il verbo RETURN (RITORNA) è utilizzato per segnalare la fine di una subroutine e comandare il ritorno nel programma.

Uso

Una subroutine può contenere più RETURN, ma solamente il primo incontrato determinerà la fine dell'esecuzione della subroutine.
La prima istruzione che sarà eseguita, dopo il RETURN, sarà quella che immediatamente segue il GOSUB o l'ON-GOSUB che ha determinato l'avvio della subroutine.
Se viene incontrato un RETURN, senza aver prima eseguito il GOSUB, s'incorre in un "ERROR 5".

Esempio

```
10 GOSUB 100
20 END
100 PRINT "CIAO"
110 RETURN
```

Quando questo programma viene eseguito, provoca la visualizzazione per una volta della parola "CIAO".

1 **STOP**

Abbreviaz.: S., ST., STO.

Vedere: END, CONT

Scopi

Il verbo STOP è utilizzato, in sede di programmazione, per fermare un programma ad un determinato passaggio per poter effettuare controlli.

Uso

Quando il verbo STOP è incontrato in un programma, l'elaborazione viene fermata e sul visore appare la parola BREAK IN..... (numero di riga dove lo STOP è collocato).
Lo STOP è utilizzato durante lo sviluppo dei programmi per controllare il funzionamento da parte di un programma o per verificare il contenuto delle variabili a quel punto.
L'esecuzione del programma potrà continuare scrivendo: CONT.

Esempio

1Ø STOP	Alla riga 1Ø si avrà uno stop che verrà visualizzato con "BREAK IN 1Ø".

1 **TROFF**

Abbreviaz.: TROF.

Vedere: TRON

Scopi

Il verbo TROFF (TRACE OFF = TRACCIA SPENTA) annulla la precedente istruzione TRON che permette di svolgere un programma seguendolo passo passo.

Uso

L'esecuzione del verbo TROFF rende nuovamente normale l'esecuzione del programma.

Esempio

```
10 TRON
20 FOR I=1 TO 3
30 NEXT I
40 TROFF
```

Quando viene eseguito questo programma, verranno visualizzati i numeri di riga 10, 20, 30, 30, 30 e 40 se è premuto il tasto [↓].

1 **TRON**

Abbreviaz.: TR., TRO.

Vedere: TROFF

Scopi

Il verbo TRON (TRACE ON = TRACCIA ACCESA) abilita l'esecuzione di un programma passo passo, visualizzando il numero di riga ed il contenuto della riga per controllare un programma.

Uso

Quando si vuole controllare un programma per verificare i movimenti del contenuto delle variabili o di salti, ecc. si può fare ricorso al verbo TRON. Quando il modo traccia è attivato, il numero di riga di ogni istruzione viene visualizzato dopo l'esecuzione di ciascuna istruzione. Il **computer** si ferma ed attende che venga premuto il tasto Freccia in basso prima di passare all'istruzione successiva. E' possibile premere il tasto Freccia in alto per vedere l'istruzione che è appena stata eseguita. Il modo traccia continua sino a che non viene eseguito il verbo TROFF o viene premuto il tasto [SHIFT] e il tasto [CA / C·CE].

Esempio

```
10 TRON
20 FOR I=1 TO 3
30 NEXT I
40 TROFF
```

Quando viene eseguito questo programma, verranno visualizzati i numeri di riga 10, 20, 30, 30, 30 e 40 se è premuto il tasto [↓].

1 **USING**
2 **USING** "specifica di revisione"

Abbreviazioni: U., US., USI., USIN.

Vedere: LPRINT, PAUSE, PRINT

Un'ulteriore guida all'uso di USING viene data nell'appendice C.

Scopi

Il verbo USING (usando) viene utilizzato per controllare il formato dei caratteri da visualizzare o da stampare.

Uso

Il verbo USING può essere usato da solo o come clausola di un'istruzione LPRINT, PAUSE o PRINT. Il verbo USING determina un formato specifico che verrà seguito sino a che non verrà cambiato da un altro verbo USING.

La specifica del verbo USING è costituita da una stringa tra virgolette formata da una combinazione dei seguenti caratteri di revisione:

Carattere di campo numerico giustificato a destra
. Punto decimale
^ Indica che il numero deve essere visualizzato in notazione scientifica
& Campo alfanumerico giustificato a sinistra

Per esempio, "####" è una specifica di revisione per un campo numerico guistificato a destra costituito da tre cifre più il segno. Nei campi numerici deve essere inclusa una posizione per il segno anche se questo è positivo.

Le specifiche di revisione possono includere più di un campo. Per esempio è possibile usare "####&&&&" per stampare un campo numerico ed un campo alfabetico l'uno dopo l'altro.

Se manca la specifica, come nella versione 1, la formattazione speciale viene disattivata e valgono le regole di visualizzazione incorporate.

Esempio

	Visore
1Ø A=125: X$="ABCDEF"	
2Ø PRINT USING "##.##^"; A	**1.25E Ø2**
3Ø PRINT USING "&&&&&&&&"; X$	**ABCDEF**
4Ø PRINT USING "####&&&"; A; X$	**125ABC**

(Per ulteriori informazioni sul funzionamento di USING consultare l'APPENDICE C)

1 **WAIT** espressione
2 **WAIT**

Abbreviaz.: W., WA., WAI.

Vedere: PRINT

Scopi

Il verbo WAIT (ATTENDI) è utilizzato per stabilire per quanto tempo un risultato permanga sul visore prima che prosegua l'esecuzione di un programma.

Uso

Normalmente un programma si ferma quando incontra un PRINT per farlo proseguire bisogna premere il tasto [ENTER].
Il verbo WAIT ordina la visualizzazione di quanto segue il verbo PRINT, per un tempo determinato, dopo di che il programma prosegue automaticamente (similmente a quanto succede utilizzando il verbo PAUSE).
L'espressione che segue il verbo WAIT stabilisce la durata della visualizzazione, che può essere regolata utilizzando un numero compreso tra Ø e 65535. Ogni incremento di 1 unità comporta un aumento di durata di 1/59 di secondo circa. Il numero Ø dà un tempo di visualizzazione troppo breve, il 59 di circa un secondo, mentre il 65535 di circa 19 minuti.
Il verbo WAIT, senza ulteriori specifiche, cancella la precedente istruzione e riporta il verbo PRINT a visualizzare normalmente i dati finché viene premuto [ENTER].

Esempio

```
1Ø WAIT 59
```

La visualizzazione del PRINT durerà circa 1 secondo.

LE FUNZIONI

Le pseudovariabili

Le pseudovariabili sono un gruppo di funzioni che non hanno argomento e vengono utilizzate come se fossero delle semplici variabili, dove occorre.

1 INKEY$

INKEY$ è una pseudovariabile stringa che assegna alla variabile specificate il valore del tasto premuto mentre viene eseguita la funzione INKEY$. I tasti [ENTER], [C-CE], [SHIFT], [DEF], [SML], [↑], [↓], [►], [◄], [CAL], [BASIC] ed i tasti delle funzioni scientifiche hanno valore NULLO. INKEY$ viene utilizzato per assegnare ad una variabile stringa un carattere senza dover poi premere il tasto [ENTER] per terminate l'inserimento.

L'esempio seguente chiarirà le idee:

```
 5 WAIT 5Ø
1Ø A$ = INKEY$
2Ø B = ASC A$
3Ø IF B = Ø THEN GOTO 1Ø
4Ø IF B...
```

Le righe da 4Ø in avanti contengono i testi da assegnare al tasto e le azioni da intraprendere. (Per esempio: 4Ø PRINT A$)

Eseguendo il programma per la prima volta, il valore di INKEY$ è nullo, perché l'ultimo tasto premuto era [ENTER].

- Se il programma dovesse iniziare con un comando INKEY$, dando inizio al programma il comando INKEY$ potrebbe leggere il segnale del tasto di inizio. Il programma seguente:

  ```
  1Ø "Z" : Z$ = INKEY$
  :
  :
  ```

  potrebbe leggere il tasto [Z] già quando si dà inizio all'esecuzione del programma, premendo i tasti [DEF] [Z].

1 **MEM**

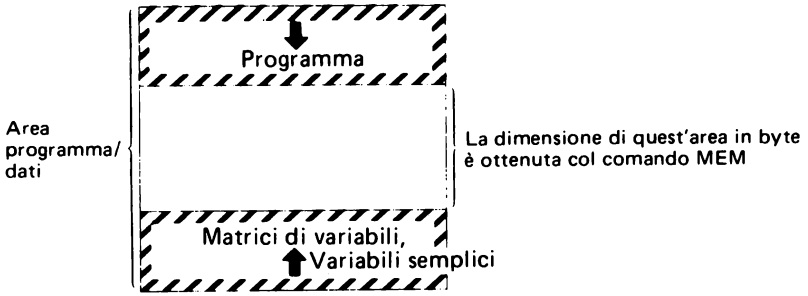
Abbreviazione: M.

Scopi

Per ottenere il numero di byte lilberi nell'area dei programmi e dei dati.

Uso

Si ottiene il numero di byte liberi (area non utilizzata dal programma, dalle matrici di variabili o dalle variabili semplici) dell'area di programma/dati.



Riferimenti

① La dimensione del programma in byte può essere ottenuta tramite la seguente operazione.

CLEAR [ENTER] (azzera le variabili semplici, le matrici di variabili, ecc).
6878 – MEM [ENTER] ← visualizza il numero di byte contenuti nel programma.

1 **PI**

PI è una pseudovariabile numerica che ha il valore del pi greco. E' indifferente l'uso del PI o del carattere speciale (π) sulla tastiera. Come gli altri numeri, il valore del PI ha una precisione contenuta in 1Ø cifre (3.141592654).

Funzioni numeriche

Le funzioni numeriche sono un gruppo di operazioni matematiche che assumono un singolo valore numerico e danno come risultato un valore numerico. Includono le funzioni trigonometriche, le funzioni logaritmiche, e le funzioni che operano sulla parte intera e sul segno di un numero. Molti dialetti del BASIC richiedono che l'argomento di una funzione sia racchiuso tra parentesi. Il **computer** non richiede queste parentesi, salvo quando è necessario indicare quale parte di un'espressione più complessa deve essere inclusa nell'argomento.

LOG 1ØØ + 1ØØ sarà interpretato dal computer come:
(LOG 1ØØ) + 1ØØ e non LOG (1ØØ + 1ØØ).

1 **ABS** (ABSOLUTE = ASSOLUTO) espressione numerica

ABS è una funzione che rende positivo il suo argomento numerico (valore assoluto di un numero = valore di un numero senza considerare il segno. ABS –1Ø = 1Ø).

1 **ACS** (ACS = ARCOCOSENO) espressione numerica

ACS è una funzione che calcola l'arcocoseno di un numero. L'arcocoseno è l'angolo il cui arcoseno è uguale all'espressione. Il valore calcolato dipenderà dal tipo di misura degli angoli selezionata (DEG–RAD–GRAD). ACS Ø.5 è 6Ø con il **computer** in DEG = gradi decimali.

1 **AHC** espressione numerica

AHC è una funzione numerica che ritorna il coseno arcoiperbolico dell'argomento numerico. AHC5 dà 2.29243167.

1 **AHS** espressione numerica

AHS è una funzione numerica che ritorna il seno arcoiperbolico dell'argomento numerico. AHS dà 2.491779853.

1 **AHT** espressione numerica

AHT è una funzione numerica che ritorna la tangente arcoiperbolica dell'argomento numerico.

1 **ASN** (ASN = ARCOSENO) espressione numerica

ASN è una funzione che calcola l'arcoseno del suo argomento. L'arcoseno è un angolo il cui seno è uguale all'espressione. Il valore calcolato dipenderà dal tipo di misura degli angoli selezionata (DEG–RAD–GRAD). ASN Ø.5 è 3Ø con il **computer** in DEG.

1 **ATN** (ATN = ARCOTANGENTE) espressione numerica

ATN è una funzione che calcola l'arcotangente del suo argomento. L'arcotangente è un angolo la cui tangente è uguale all'espressione. Il valore calcolato dipenderà dal tipo di misura degli angoli, selezionata (DEG–RAD–GRAD). ATN 1 è 45 in DEG.

1 **COS** espressione numerica

COS è una funzione numerica che ritorna il coseno dell'argomento dell'angolo. Il valore ritornato varia a seconda che il computer sia nel modo a gradi decimali, a radianti o a gradi. COS 60 dà 0.5, nel modo a gradi decimali.

1 **CUR** espressione numerica

CUR è una funzione numerica che ritorna la radice cubica del suo argomento. CUR 8 dà 2.

1 **DEG** (DEG = DEGREE = GRADI DECIMALI) espressione numerica

La funzione DEG converte il valore dell'angolo in argomento espresso in DMS (DEGREE – MINUTE – SECOND = GRADI – MINUTI – SECONDI) in DEG. In DMS la parte intera rappresenta i gradi, il primo e il secondo decimale rappresentano i minuti, il terzo e quarto decimale rappresentano i secondi. Ulteriori decimali rappresentano decimi di secondo. Ad esempio 55°1Ø'44.5'' è scritto

55.1Ø445. Con la rappresentazione in DEG la parte intera rappresenta i gradi mentre la parte decimale rappresenta i gradi decimali.
DEG 55.1Ø445 à 55.17ØØ2778.

1 **DMS** (DMS = DEGREE – MINUTE – SECOND = GRADI-MINUTI-SECONDI) espressione numerica

DMS è una funzione numerica che converte il valore dell'angolo in argomento dal formato DEG (gradi decimali) al formato DMS (gradi minuti secondi).
DMS 55.17ØØ2778 è 55.1Ø445.

1 **EXP** (EXP = ESPONENZIALE) espressione numerica

EXP è una funzione numerica corrispondente alla funzione inversa del logaritmo naturale. EXP 1 è 2.718281828.

1 **FACT** espressione numerica

FACT è una funzione numerica che ritorna il fattoriale del suo argomento. FACT 5 dà 120.

1 **HCS** espressione numerica

HCS è una funzione numerica che ritorna il coseno iperbolico dell'argomento numerico. HCS 5 dà 74.20995852.

1 **HSN** espressione numerica

HSN è una funzione numerica che ritorna il seno iperbolico dell'argomento numerico.

1 **HTN** espressione numerica

HTN è una funzione numerica che ritorna la tangente iperbolica dell'argomento numerico. HTN 1 è 0.761594156.

1 **INT** (INT = INTERO) espressione numerica

INT è una funzione numerica che calcola il valore intero di un numero. INT PI è 3.

1 **LN** (LN = LOGARITMO NATURALE) espressione numerica

LN è una funzione numerica che calcola il logaritmo naturale in base e (2.718281828) del suo argomento numerico. LN 1ØØ è 4.6Ø5171Ø186.

1 **LOG** (LOG = LOGARITMO) espressione numerica

LOG è una funzione numerica che calcola il logaritmo in base 1Ø del suo argomento. LOG 1ØØ è 2.

1 **POL** (espressione numerica, espressione numerica)

POL è una funzione numerica che converte gli argomenti numerici di coordinate rettangolari in argomenti di coordinate polari.
Il primo argomento numerico indica la distanza dall'asse y e il secondo la distanza dall'asse x. I valori convertiti, la distanza e l'angolo nelle coordinate poolario vengono asegnati alle variabili fisse Y e Z, rispettivamente. L'angolo convertit è diverso a seconda che il computer sia predisposto nel modo a gradi decimali, a radianti o a gradianti. POL (3, 4) dà (5, 53.13010235), in gradi decimali.

1 **RCP** espressione numerica

RCP è una funzione numerica che ritorna il reciproco del suo agromento. RCP 5 dà 0.2

1 **REC** (espressione numerica, espressione numerica)

REC è una funzione numerica che converte gli argomenti numerici dal formato in coordinate polari al formato in coordinate rettangolari.

Il primo argomento numerico indica la distanza e il secondo argomento numerico l'angolo, il quale varia a seconda che il computer sia predisposto per il modo a gradi decimali, a radianti, o a gradianti. I valori convertiti e le distanze dall'asse x e y vengono assegnati alle variabili fisse Y e Z rispettivamente. REC (6, 50) dà (4.400513268, 5.362311102).

1 **RND** (RND = RANDOM = A CASO) espressione numerica

RND è una funzione numerica che genera numeri a caso. Se il valore dell'argomento è inferiore a 1 ma maggiore od uguale a zero, il numero generato sarà inferiore a 1 e maggiore od uguale a zero. Se il suo argomento è un intero maggiore od uguale ad 1, il risultato sarà un numero generato a caso maggiore od uguale ad 1 ed inferiore od uguale all'argomento.
Se l'argomento è maggiore od uguale a zero e on intero, il risultato è un numero casuale maggiore od usuale ad 1 ed inferiore od unguale al più piccolo intero maggiore dell'argomento.

	Risultato	
ARGOMENTO	LIMITE INFERIORE	LIMITE SUPERIORE
.5	Ø<	<1
2	1	2
2.5	1	3

Viene normalmente generata la stessa sequenza di numeri casuali in quanto viene utilizzato sempre lo stesso numero di partenza. Per casualizzare anche il numero di partenza vedere il verbo RANDOM.

1 espressione numerica **ROT** espressione numerica

ROT è una funzione numerica che ritorna la radice del suo argomento.
125 ROT 3 dà 5.
(V dovrebbe essere impostato con la scrittura: 125 ROT 3).

1 **SGN** (SGN = SEGNO) espressione numerica

SGN è una funzione numerica che calcola un valore basandosi sul segno dell'argomento. Se l'argomento è positivo, il risultato è 1. Se l'argomento è zero il risultato sarà Ø, se l'argomento è negativo, il risultato sarà −1. SGN −5 è −1.

1 **SIN** (SIN = SEGNO) espressione numerica

SIN è una funzione numerica che calcola il seno dell'angolo in argomento. Il valore calcolato dipende dai tipi di misura degli angoli impostato sul **computer** (DEG–RAD–GRAD). SIN 3Ø è Ø.5 in DEG.

1 **SQR** (SQR = SQUARE = QUADRATO) espressione numerica

SQR è una funzione numerica che calcola la radice quadrata del suo argomento. L'uso è identico a quello del suo simbolo ($\sqrt{\ }$) posto sulla tastiera. SQR 4 è 2.

1 **SQU** espressione numerica

SQU è una funzione numerica che ritorna il quadrato del suo argomento numerico. SQU 3 dà 9.

1 **TAN** (TAN = TANGENTE) espressione numerica

TAN è una funzione numerica che calcola la tangente dell'angolo in argomento. Il valore calcolato dipende dal tipo di misura degli angoli prescelto (DEG–RAD–GRAD). TAN 45 è 1 in DEG.

1 **TEN** espressione numerica

TEN è una funzione numerica che ritorna il valore di 10 (la base del logaritmo comune) elevato al valore del suo argomento numerico. TEN 3 dà 1000.

Funzioni stringa

Le funzioni stringa sono un gruppo di operazioni usate per manipolare le stringhe. Alcune di queste funzioni prendono un argomento stringa e calcolano un valore numerico. Alcune prendono un argomento stringa e calcolano una stringa. Alcune prendono un valore numerico e calcolano una stringa. Alcune prendono un argomento stringa ed uno o due argomenti numerici e calcolano una stringa. Molti dialetti del BASIC richiedono che l'argomento di una funzione sia messo tra parentesi. Il **computer** non richiede queste parentesi, eccetto quando è necessario indicare quale parte di un'espressione più complessa deve essere inclusa nell'argomento. Le funzioni stringa con due o tre argomenti richiedono tutte le parentesi.

1 **ASC** (ASCII) espressione stringa

La funzione stringa ASC converte il primo carattere in argomento nel corrispondente codice carattere numerico. La tabella dei codici per i caratteri e le loro relazioni coi caratteri sono illustrate nell'appendice B.
Esempio: ASC "A" [ENTER] → 65.
Il PC-1350 utilizza i caratteri ASCII e i loro codici.

1 **CHR$** (CHR = CHARACTER = CARATTERE) espressione numerica

La funzione stringa CHR$ converte l'argomento numerico nel codice carattere corrispondente. La tabella dei codici per i caratteri e le loro relazioni coi caratteri sono illustrate nell'appendice B.
Esempio: CHR$ 65 [ENTER] → "A".

1 **LEFT$** (LEFT = SINISTRA) espressione stringa, espressione numerica

La funzione stringa LEFT$ permette di leggere dall'espressione stringa in argomento il numero di caratteri, partendo da sinistra, indicato nell'espressione numerica in argomento.
Esempio: LEFT$ ("ABCDEF", 2) [ENTER] → "AB".

1 **LEN** (LEN = LENGTH = LUNGHEZZA) espressione stringa

La funzione stringa LEN calcola la lunghezza della stringa in argomento.
Esempio: LEN "ABCDEF" [ENTER] → 6.

1 **MID$** (MID = MIDDLE = MEZZO) espressione stringa, esp. numerica 1, esp. numerica 2

MID$ è una funzione stringa che ritorna la parte centrale della stringa del primo argomento. Il primo argomento numerico indica la posizione del primo carattere da includere nel risultato. Il secondo argomento numerico indica il numero di caratteri da includere.
MID$ ("ABCDEF", 2, 3) [ENTER] → "BCD".

1 **RIGHT$** (RIGHT = DESTRA) espressione stringa, espressione numerica

La funzione stringa RIGHT$ permette di leggere i caratteri dell'espressione stringa in argomento, partendo da destra, nella quantità indicata nell'espressione numerica in argomento.
Esempio: RIGHT$ ("ABCDEF", 3) [ENTER] → "DEF".

1 **STR$** (STR = STRING = STRINGA) espressione numerica

La funzione STR$ permette di convertire l'espressione numerica in argomento nella corrispondente stringa. È l'inverso di VAL.
Esempio: STR$ 1.59 [ENTER] → "1.59".

1 **VAL** (VAL = VALUE = VALORE) espressione stringa

L'espressione stringa VAL permette di convertire l'espressione stringa in argomento, nella corrispondente espressione numerica.
È l'inverso di STR$.
La funzione VAL di un NULLO è zero.
Esempio: VAL "1.59" [ENTER] → 1.59.

NOTA: La stringa di caratteri convertibile dalla funzione VAL in valore numerico consiste di numeri (da Ø a 9), dei simboli + e –, e dal simbolo E della parte esponenziale. Non può essere usato nessun altro carattere o simbolo. Se la stringa di caratteri dovesse includere altri caratteri o simboli, qualsiasi stringa di caratteri a destra di quella stringa di caratteri viene ignorata. Se la stringa di caratteri dovesse contenere uno spazio, quest'ultimo solitamente viene ignorato.

CAPITOLO 9
RICERCA DEI GUASTI

Questo capitolo Vi insegna cosa fare quando il Vostro **computer** non fa quello che vi aspettate faccia. Il capitolo è diviso in due parti, la prima tratta la macchina in generale, la seconda tratta la programmazione in Basic. Per ogni problema possono esistere più soluzioni. Provatele una alla volta fino ad ottenere la risoluzione del problema.

Problemi con la macchina

Accendete il computer ma non appare nulla sul visore.	1) Controllate che l'interruttore sia su ON. 2) Premete il tasto [ON BRK] per controllare se l'autospegnimento era attivato. 3) Sostituite le pile. 4) Regolate il controllo del contrasto.
Il visore è acceso ma i tasti non funzionano.	1) Premete il tasto [C·CE] . 2) Premete [CA] ([SHIFT] [C·CE]). 3) Spegnete e riaccendete il computer. 4) Tenete premuto un qualsiasi tasto alfanumerico e premete il pulsante RESET. 5) Premete solo il pulsante RESET.
Avete inserito un calcolo o attendete una risposta in un programma ma non appare nulla.	1) Premete [ENTER] .
State elaborando un programma. Viene visualizzato qualche cosa. Il programma non prosegue.	1) Premete [ENTER] .
Avete inserito un calcolo e viene visualizzato come istruzione in Basic (due punti dopo il primo numero).	1) Passate dal modo PROgramma al modo RUN per i calcoli.

Non riuscite ad ottenere il funzionamento dei tasti.	1) Tenete premuto un tasto e contemporaneamente premete il pulsante RESET. 2) Se, dopo avere eseguito l'operazione descritta, nessuno dei tasti è operativo, premete il tasto RESET. Quindi, premete il tasto [Y], quello [=] o quello [ENTER] in risposta al messaggio "MEMORY CLEAR O.K.?". I programmi ed i dati in memoria verranno cancellati.

Controllo dei programmi

Quando si memorizza un nuovo programma è normale che la prima volta che lo si fa "girare" non funzioni. Questo succede anche quando il programma è semplice e Voi siete sicuri del suo funzionamento. Infatti è normale sbagliare premendo un tasto al posto di un altro. Per aiutarVi a scoprire e correggere questi errori, Vi diamo qui di seguito alcuni suggerimenti:

a) Siete incorsi in un errore durante l'elaborazione di un programma.

1. Tornate in PRO ed utilizzando i tasti [↑] e [↓] richiamate la riga che contiene l'errore. Il cursore si troverà posizionato nel punto dove il **computer** è andato in confusione.

2. Se secondo Voi l'errore non è giustificato in quanto la riga è corretta, quasi sicuramente il problema sarà legato ai valori usati. Ad esempio $\sqrt{A}$ produrrà errore se A contiene −1. Controllate il contenuto delle variabili, sia in PRO scrivendo il nome della variabile desiderato e facendolo seguire dal tasto [ENTER].

b) Elaborate un programma, tutto funziona regolarmente ma non otterrete il risultato atteso.

3. Controllate le righe del programma usando LIST ed i tasti [↓] e [↑] per vedere se tutto è stato inserito correttamente (una riga al posto di un'altra). E' sorprendente vedere quanti errori si possono scoprire semplicemente ricontrollando il programma.

4. Pensate come il computer! Seguite il programma riga per riga, e nell'ambito di ogni riga, istruzione per istruzione. Prendete dei valori facilmente "maneggiabili" ed inseriteli nelle variabili per osservarne il comportamento. Provate ad eseguire il calcolo, riportato in una riga, su di un pezzo di carta. Attenti alle parentesi.

5. Aggiungete dei PRINT extra per verificare i calcoli in differenti posizioni di programma. In questa maniera riuscirete ad isolare la parte di programma che non funziona come dovrebbe. Questo sistema è utile anche per determinare quali parti del programma sono state eseguite. Potete anche usare lo STOP per fermare il programma nei suoi punti critici e controllare i contenuti delle variabili.

6. Usate il TRON ed il TROFF per seguire lo svolgimento del programma passo passo. FermateVi a controllare il contenuto delle variabili nei passaggi cruciali. Questo sarà un metodo lento per risolvere un problema, ma spesso sarà anche l'unica strada possibile.

CAPITOLO 10
MANUTENZIONE

Per evitare d'incorrere nel malfunzionamento del Vostro **computer** Vi suggeriamo quanto segue:

* Maneggiate sempre con cura il Vostro **computer**. Il visore a cristalli liquidi è particolarmente fragile.
* Tenete il computer lontano da fonti di calore e non sottoponetelo a repentini sbalzi di temperatura. Tenetelo lontano dall'umido e dalla polvere. Durante l'estate non lasciate il **computer** in macchina od esposto direttamente ai raggi solari.
* Usate solo un panno morbido per pulire il **computer**. Non usate solventi, acqua o panni umidi.
* Per evitare che le pile possono deteriorarsi, all'interno del computer, toglietele se prevedete di non usarlo per lungo tempo.
* Per qualsiasi necessità tecnica, rivolgersi esclusivamente ad un rivenditore SHARP, ad un centro SHARP o ad un servizio riparazioni SHARP autorizzato.
* Se il computer è sottoposto a forti scariche elettrostatiche o a fortissimi rumori, può bloccarsi. In questo caso bisognerà premere il pulsante RESET. Unitamente ad un altro tasto (vesi "ricerca dei guasti.)
* Conservate il manuale e consultatelo nel dubbio.

APPENDICE A
MESSAGGI DI ERRORE

Il **computer** prevede nove diversi codici di errore. Questi codici verranno spiegati nella tabella seguente.

Numero di errore	Significato
1	Errore di sintassi.

- Significa che il **computer** non riesce a capire le istruzioni che gli avete dato. Controllate particolari quali il punto a virgola alla fine delle istruzioni PRINT, le parole scritte in mode errato, e gli usi non corretti.

  3 * / 2

2 Errore di calcolo

Probabilmente avete fatto una delle tre cose seguenti:

1. Avete cercato di usare un numero troppo grande.
   I risultati dei calcoli sono superiori a 9.999999999E 99.
2. Avete cercato di dividere per zero.

   5 / Ø

3. E'stato tentato un calcolo illogico.

   LN-3Ø o ASN 1.5

3 Errore di DIMensionamento/errore di argomento.

- La matrice esiste già.

  La matrice è stata specificata senza prima dimensionarla.

  Il suffisso della matrice supera la dimensione di matrice specificata nell'istruzione DIM.

  DIM B (256)

- L'argomento della funzione è illecito. Ciò significa che avete cercato di far fare al computer qualcosa che non è in grado di fare.
  Intervallo superiore a 65535.

  WAIT 66ØØØ

4 Errore di riga

In questo caso avete probabilmente fatto una delle due cose seguenti:

1. Cercato di usare un numero di riga inesistente con l'istruzione GOTO, GOSUB, RUN, LIST o THEN etc.
2. Cercato di usare un numero di riga troppo grande. Il numero di riga massimo è 65279.

5 Errore di intercalazione

L'intercalazione delle subroutine supera 10 livelli.

L'intercalazione del ciclo FOR supera 5 livelli.

Verbo RETURN senza un GOSUB, verbo NEXT senza un FOR o verbo READ senza un DATA.

E'stato superato lo spazio della memoria di transito (buffer).

6 Superamento della capacità della memoria

Generalmente questo errore si verifica quando cercate di dimensionare una matrice troppo grande per la memoria. Può verificarsi inoltre quando un programma diventa troppo lungo.

7 Errore PRINT USING

Significa che avete specificato un formato illecito nell'istruzione USING.

8 Errore del dispositivo I/O

Questo errore può verificarsi soltanto quando avete collegato al **computer** la stampante e/o il registratore a cassette opzionali. Significa che vi è un problema di comunicazione tra il dispositivo I/O ed il **computer**.

9 Altri errori

Questo codice verrà visualizzato ogniqualvolta il computer ha un problema che non rientra in uno degli altri codici di errore. Una delle cause più comuni di questo errore consiste nel cercare di accedere ai dati contenuti in una variabile in un modo (per esempio A$) mentre i dati sono stati memorizzati in origine nella variabile in un altro modo (per esempio A).

ERRORI LEGATI A RENUM

Messaggio di errore	Descrizione
ERROR 1	E stato commesso un errore di sintassi nel comando RENUM.
ERROR 1 numero di riga	Nel comando che specifica la destinazione del salto manca il numero di riga (ad es. GOTO, GOSUB ecc.)
ERROR 3 numero di riga	Durante l'esecuzione del comando RENUM è stato incontrato un numero di linea superiore a 65279. La lunghezza di una riga di programma supera i 79 byte.
ERROR 4	Il vecchio numero di riga specificato non esiste nel programma.
ERROR 4 numero di riga	Il numero di riga apecificato come destinazione del salto non esiste nel programma.
ERROR 6	La capacità della memoria è insufficiente per eseguire il comando RENUM o diventa scarsa durante il processo di renumerazione.
ERROR 9	Si è tentato di esguire il comando RENUM in un modo diverso da quello PRO (programmazione). Si è tentato di cambiare l'ordine di esecuzione delle righe di programma specificando un nuovo numero di riga più basso del numero immediatamente precedente il vecchio numero di riga.
ERROR 9 numero di riga	Il numero di riga specificato come destinazione del salto è inappropriato perchè impiega una variabile, un'espressione o una funzione (ciòe un riferimento numerico di riga scorretto).

APPENDICE B
TAVOLA DEI CODICI ASCII

La seguente tabella dà i valori di conversione da usarsi con CHR$ e ASC.

Le colonne indicano il primo carattere esadecimale o i primi quattro numeri del codice binario. Le righe indicano il secondo carattere esadecimale o le seconde quattro cifre del codice binario. L'angolo superiore di sinistra di ogni quadratino contiene il numero decimale di rappresentazione del carattere. L'angolo inferiore destro contiene il carattere. Se non vi sono dei caratteri significa che si tratta di un carattere non accettato dal **computer.**

Ad esempio, il carattere "A" è il numero decimale 65, esadecimale 41, o il codice binario 01000001. Il carattere "√" è il numero decimale 252, esadecimale FC o il codice binario 11111100.

Note:
- I caratteri per i codici di caratteri 92 (&5C), 249 (&F9) e 250 (&FA) che appaiono nel display del computer sono diversi dai caratteri stampati con le stampanti opzionali CE-126P in corrispondenza dei medesimi codici.
- Il carattere stampato in corrispondenza del codice 92 (&5C), elencato nella tabella seguente, è ◣ con la stampante CE-126P.
- Se si usa la stampante CE-126P, non usare il codice 0 (&00).
- Qualsiasi codice da 0 (&00) a 31 (&1F) non usato per dei caratteri produce la stampa di uno spazio.
- I codici 249 (&F9) e 250 (&FA) sono spazi.

Tabella dei codici ASCII

Primi 4 bit

Secondi 4 bit

Hex Binary	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	E 1110	F 1111
0 0000	0 NUL	16	32 SPACE	48 0	64 @	80 P	96 `	112 p	128	224	240
1 0001	1	17	33 !	49 1	65 A	81 Q	97 a	113 q	129	225	241
2 0010	2	18	34 "	50 2	66 B	82 R	98 b	114 r	130	226	242
3 0011	3	19	35 #	51 3	67 C	83 S	99 c	115 s	131	227	243
4 0100	4	20	36 $	52 4	68 D	84 T	100 d	116 t	132	228	244
5 0101	5	21	37 %	53 5	69 E	85 U	101 e	117 u	133	229	245 ♠
6 0110	6	22	38 &	54 6	70 F	86 V	102 f	118 v	134	230	246 ♥
7 0111	7	23	39	55 7	71 G	87 W	103 g	119 w	135	231	247 ♦
8 1000	8	24	40 (	56 8	72 H	88 X	104 h	120 x	136	232	248 ♣
9 1001	9	25	41)	57 9	73 I	89 Y	105 i	121 y	137	233	249 ▮
A 1010	10	26	42 *	58 :	74 J	90 Z	106 j	122 z	138	234	250 ⌐
B 1011	11	27	43 +	59 ;	75 K	91 [	107 k	123 {	139	235	251 π
C 1100	12	28	44 ,	60 <	76 L	92 \	108 l	124 |	140	236	252 √
D 1101	13	29	45 –	61 =	77 M	93]	109 m	125 }	141	237	253
E 1110	14	30	46 .	62 >	78 N	94 ^	110 n	126 ~	142	238	254
F 1111	15	31	47 /	63 ?	79 O	95 _	111 o	127	143	239	255

APPENDICE C
I FORMATI DI VISUALIZZAZIONE

Alcune volte è importante o utile poter controllare il formato ed il contenuto dei risultati. Grazie al verbo USING queste operazioni sono possibili e consentono di specificare:

* Numero delle cifre
* Posizione del punto decimale
* Formato della notazione scientifica
* Numero dei caratteri della stringa

I risultati che si vogliono visualizzare con queste specifiche sono definiti per mezzo di una "Maschera" che può essere una stringa costante o variabile: costante, quando il formato specificato è fisso:

```
10 USING "####"
```

Variabile, quando il formato è definito da una variabile stringa.

```
20 M$ = "&&&&&&"
30 USING M$
```

Quando il verbo USING è usato senza la maschera di definizione, cancella tutti i precedenti formati.

```
40 USING
```

Il verbo USING può essere utilizzato in unione al verbo PRINT.

```
50 PRINT USING M$; N
```

L'uso del verbo USING influenza tutti i successivi PRINT che verranno controllati secondo il formato specificato fino a che non verrà incontrato un nuovo USING.

Maschere numeriche

Le maschere numeriche USING possono essere usate solo per visualizzare dei valori numerici, cioé costanti numeriche o variabili numeriche. Se viene visualizzata una stringa costante o variabile mentre è in atto una maschera USING numerica, la maschera verrà ignorata. Il valore che deve essere visualizzato deve sempre essere adatto allo spazio previsto dalla maschera. La maschera deve riservare lo spazio per il segno, anche quando il numero sarà sempre positivo. Pertanto una maschera che indica quattro posizioni da visualizzare potrà essere usata soltanto per visualizzare i numeri costituiti da tre cifre.

Specifica del numero delle cifre

Il numero desiderato di cifre viene specificato usando il carattere "#". Ogni "#" che compare nella maschera riserva lo spazio per una cifra. La visualizzazione o la stampa contengono sempre tanti caratteri quanti sono stati indicati nella maschera. Il numero compare all'estrema destra di questo campo; le posizioni restanti a sinistra vengono riempite con degli spazi. I numeri positivi pertanto hanno sempre almeno uno spazio alla sinistra del campo. Poiché **computer** può accettare numeri aventi un massimo di 10 cifre significative, nelle maschere numeriche non si devono usare più di 11 "#".
Quando il numero totale di colonne della parte intera specificata supera 11, questa parte intera viene considerata come costituita da 11 cifre nel **computer**.

Nota: In tutti gli esempi di questa appendice, l'inizio e la fine del campo visualizzato saranno indicati da un carattere "|" ad indicare la dimensione del campo.

Istruzione	Visore
10: USING "####"	(Mettete il **computer** nel modo RUN; battete RUN e premete [ENTER])
20: PRINT 25	| 2 5|
30: PRINT −350	|− 3 5 0|
40: PRINT 1000	ERROR 7 IN 40

Notate che l'ultima istruzione ha prodotto un errore in quanto erano necessarie 5 posizioni (4 cifre e uno spazio per il segno) mentre nella maschera ne erano previste 4.

Specifica del punto decimale

Il punto decimale "." può essere incluso in un maschera per indicare il tipo di visualizzazione desiderato. Mentre la parte intera di un numero deve essere rappresentata nella sua completezza, la parte decimale può essere rappresentata come meglio si desidera. Pertanto se vogliamo la visualizzazione di un numero con solo 2 decimali, quelli in eccesso verranno troncati (non arrotondati) mentre se abbiamo più posizioni nei decimali rispetto all'effettivo valore numerico, le cifre mancanti saranno completate con degli zero.

Esempio	Visore
10: USING "####.##"	
20: PRINT 25	| 25.00|
30: PRINT −350.5	|−350.50|
40: PRINT 2.547	| 2.54|

Specifica della notazione scientifica

Il simbolo "^" indica che il numero deve essere visualizzato o stampato in notazione scientifica. Il simbolo "#" e "." sono usati per specificare le caratteristiche del numero ovvero, delle cifre poste a sinistra del simbolo **E**. Due "##" devono essere sempre messi alla sinistra del punto decimale, per il segno il punto decimale può essere messo ma non é determinante. Si possono visualizzare fino a 9 cifre alla destra del punto decimale. Il computer infine visualizzerà il numero, come da noi specificato con il simbolo di notazione scientifica (**E**), seguito da uno spazio per il segno più ed altri due per l'esponente. Pertanto volendo specificare una maschera che ci rappresenti un numero in notazione scientifica scriveremo: "##^", ed otterremo una visualizzazione del tipo **2E** 99. Volendo ottenere la rappresentazione del campo più grande possibile scriveremo una maschera "##.#########^" che visualizzerà un numero, ad esempio, così composto: −1.2345678900 **E** −12

Esempio	Visore
10: USING "###.##^"	
20: PRINT 2	| 2.00E 00|
30: PRINT −365.278	|−3.65E 02|

Specifica della maschera alfanumerica

Per visualizzare delle stringhe o dei caratteri in determinati formati, si usa, nelle specifiche della maschera, il simbolo "&". Ognuno di questi simboli specifica un carattere. La stringa sarà posizionata a partire dal limite sinistro del visore. Se la stringa è più corta del campo specificato dalla maschera, gli spazi in eccesso saranno completati con degli spazi, se più lunga, verrà troncata.

Esempio	Visore
1Ø: USING "&&&&&&"	
2Ø: PRINT "ABC"	|A B C |
3Ø: PRINT "ABCDEFGHI"	|ABCDEF|

Maschere miste

Nella maggior parte delle applicazioni, le maschere USING conterranno tutti numeri o tutte stringhe. E' tuttavia possibile per alcune applicazioni, includere in una sola maschera USING entrambi i tipi di caratteri. In tali casi, ogni passaggio dai caratteri numerici ai caratteri di stringa o viceversa, delimita un valore diverso. Pertanto, una maschera del tipo "#####&&&&" specifica la visualizzazione di due valori separati – un valore numerico a cui sono assegnate 5 posizioni ed un valore di stringa a cui sono assegnate 4 posizioni:

Istruzione	Visore
1Ø: PRINT USING "###.##&&"; 25; "CR"	| 25. ØØCR|
2Ø: PRINT –5.789; "DB"	| –5. 78DB|

Ricordatevi che quando viene specificato un formato di visualizzazione, questo sarà valido per tutti gli output successivi sino a che un ulteriore USING cambi il formato o cancelli l'istruzione.

APPENDICE D
VALUTAZIONE DELLE ESPRESSIONI E PRIORITA DEGLI OPERATORI

Quando nel **computer** viene impostata un'espressione complessa, esso valuta le parti dell'espressione in una sequenza determinata dalla priorità delle singole parti dell'espressione. Se scrivete l'espressione:

100 / 5+ 45

sia come calcolo che come parte di un programma, il **computer** non sa se volevate intendere:

$$\frac{100}{5+45} = 2 \qquad \text{oppure} \qquad \frac{100}{5} + 45 = 65$$

Poiché il **computer** deve avere il modo di decidere tra queste opzioni, esso usa le proprie regole di priorità degli operatori. Poiché la divisione ha una "priorità" superiore all'addizione (vedi sotto), sceglierà di eseguire per prima la divisione e quindi l'addizione, cioé sceglierà la seconda opzione e darà un risultato di 65.

Priorità degli operatori

Gli operatori nel modo BASIC del **computer** vengono valutati con le seguenti priorità in ordine decrescente:

Operazioni di livello 1

1. Parentesi
2. Variabili e pseudovariabili
3. Funzioni
4. Esponenti (^), (ROT)
5. Meno unario, segno negativo (–)
6. Moltiplicazione e divisione (* , /)
7. Addizione e sottrazione (+, –)
8. Operatori di confronto (<, <=, =, <>, >=, >)
9. Operatori logici (AND, OR, NOT, XOR)

Quando vi sono due o più operatori con lo stesso grado di priorità, l'espressione verrà risolta partendo da sinistra verso destra, (gli esponenti verranno risolti da destra verso sinistra).

Nel caso di A+B–C, per esempio, il risultato è uguale sia che venga eseguita per prima la sottrazione che l'addizione.

Quando un'espressione contiene diverse parentesi le une all'interno delle altre, la serie più interna viene risolta per prima e la risoluzione procede quindi verso l'esterno.

Per i livelli 3 e 4, l'ultimo dato immesso ha la priorità superiore:

Esempio: $-2 \wedge 4 \rightarrow -(2^4)$
$3 \wedge -2 \rightarrow 3^{-2}$

Esempi di valutazione delle priorità

Partendo da questo esempio

((3 + 5 – 2) * 6 + 2) / 1Ø ^ LOG 1ØØ

Il **computer** valuterà per primo il calcolo contenuto nella parentesi più interna essendoci due operatori con la stessa priorità (+ e –) effettuerà il calcolo partendo da sinistra verso destra per cui l'addizione sarà la prima ad essere eseguita.

((8 – 2) *6 + 2) / 1Ø ^ LOG 1ØØ

Quindi viene effettuata la sottrazione e si risolve così la parentesi più interna.

(6 * 6 + 2) / 1Ø ^ LOG 1ØØ

In questa parentesi la moltiplicazione ha la priorità sull'addizione

(36 + 2) / 1Ø ^ LOG 1ØØ

quindi viene eseguita l'addizione

38 / 1Ø ^ LOG 1ØØ

ora che le parentesi sono risolte, il più alto livello di priorità è assegnato alla funzione LOG per cui si ha

38 / 1Ø ^ 2

ora calcola la potenza

38 / 1ØØ

ed infine viene eseguita la divisione.

Ø,38

Questo è il risultato dell'espressione.

APPENDICE E
FUNZIONI DEI TASTI NEL BASIC

ON / **BRK**

(ON)
Tasto di accensione.
(BREAK)

- Durante l'esecuzione di un programma, questo tasto funziona come un tasto BREAK e la sua pressione provoca l'interruzione dell'esecuzione del programma.
- Quando viene premuto durante l'esecuzione manuale, comandi di immissione/emissione quali BEEP, CLOAD, etc., vengono interrotti.

SHIFT

- Il tasto giallo contrassegnato "SHIFT" deve essere usato per accedere ad una seconda funzione. (Quanto indicato in marrone sopra a ciascun tasto.)

  Es. SHIFT [?/U] → viene immesso ?

C·CE

- Questo tasto viene usato per cancellare i dati immessi ed il visore. (Sblocco dell'errore)

SHIFT **CA**

- Non soltanto cancella il visore, ma ripristina il computer alla sua condizione iniziale.

  – Condizione iniziale –
  - Ripristina il temporizzatore WAIT
  - Ripristina il formato di visualizzazione (formato USING)
  - Ripristina la condizione TRON (TROFF)
  - Ripristina la condizione PRINT = LPRINT
  - Azzera l'errore.

0 ~ **9**

- Tasti numerici

•

- Punto decimale.
- Viene usato per immettere un'abbreviazione di un'istruzione/comando.
- Viene usato per indicare la parte decimale nell'istruzione di formato USING.

E

- Usare questo tasto della lettera E per designare un esponente, usando la scrittura scientifica.

EXP

- Viene usato per indicare un esponente in notazione scientifica.

/

- Divisione

[*]
- Moltiplicazione
- Usato per designare un variabile di matrice in INPUT#, PRINT #, ecc.

[+]
- Addizione

[−]
- Sottrazione

[SHIFT] [?]
- Viene usato per dare il comando CLOAD?

[SHIFT] [:]
- Viene usato per dividere due o più istruzioni in una riga.

[,]
- Viene usato per dare una pausa tra due equazioni e tra le variabili o i commenti.

[SHIFT] [;]
- Viene usato per ottenere una visualizzazione multipla (due o più valori/contenuti visualizzati contemporaneamente).
- Viene usato per dare una pausa tra l'istruzione e la variabile.

[=]
- Nelle istruzioni di assegnazione, viene usato per assegnare il contenuto (numeri o lettere) a destra alla variabile specificata a sinistra.
- Viene usato immettendo gli operatori logici nella frase IF.

[DEF]
- Quando viene premuto uno qualsiasi dei 18 tasti alfabetici (A, S, D, F, G, H, J, K, L, Z, X, C, V, B, N, M, ', spazio) dopo aver premuto il tasto [DEF], si dà inizio all'esecuzione del programma dalla riga di programma che ha la stessa etichetta del tasto premuto.

[A] ~ [Z]
- Tasti alfabetici

[(], [)]
- Vengono usati per immettere le parentesi.

[SHIFT] [<] / [SHIFT] [>]
- Vengono usati per immettere gli operatori logici.

[SPC]
- Viene usato per inserire uno spazio quando si immettono i programmi o i caratteri. Lo spazio viene ignorato nell'esecuzione dei programmi.

[SHIFT] [^]
- Viene usato per le istruzioni del calcolo delle potenze.
- Viene usato per specificare il sistema con punto decimale mobile (visualizzazione dell'esponente) per i dati numerici nelle istruzioni USING.

[SHIFT] [π]
- Viene usato per indicare il pi greco (π).

[√]
- Viene usato per indicare la radice quadrata.

APPENDICE E
Funzioni dei tasti nel BASIC

[SHIFT] [!] ["] [#] [$] [%] [&] [SHIFT] [@]

- Designa solo il simbolo. Ha quindi esclusivamente funzioni grafiche.
  - " : • Viene usato per indicare ed annullare i caratteri.
    • Viene usato per specificare le etichette.
  - # : Viene usato con l'istruzione USING per dare le istruzioni per definire il formato di visualizzazione dei dati numerici.
  - $: Viene usato per assegnare le variabili alfabetiche.
  - & : • Viene usato con l'istruzione USING per dare le istruzioni per definire il formato di visualizzazione delle stringhe di caratteri.
    • Viene usato per indicare un numero esadecimale.

[►]

- Sposta il cursore a destra (premendolo una sola volta si avanza di una posizione, tenendolo premuto si ottiene l'avanzamento automatico).
- Esegue le istruzioni di riproduzione.
- Annulla una condizione di errore nel funcionamento manuale.

[◄]

- Sposta il cursore a sinistra (premendolo una sola volta si avanza di una posizione, tenendolo premuto si ottiene l'avanzamento automatico).
- Esegue le istruzioni di riproduzione.
- Annulla una condizione di errore nel funzionamento manuale.

[SHIFT] [INS]

- Inserisce uno spazio (compare ⎵) della capacità di un passo tra l'indirizzo (N) indicato dal cursore e l'indirizzo precedente (N-1).

[SHIFT] [DEL]

- Cancella il contenuto dell'indirizzo (N) indicato dal cursore.

[SHIFT] [INPUT] ~ [SHIFT] [CLOAD]

- Tasti di comando e di istruzione prestabiliti. Premendo [SHIFT] e quindi il tasto alfabetico (incluso il segno di uguale, lo spazio e [ENTER]) sotto al comando ed all'istruzione desiderati, si dà il comando e l'istruzione desiderati al **computer**.

[CAL]

- Viene usato per impostare il modo CAL.

[BASIC]

- Viene usato per passare al modo RUN quando è impostato il modo CAL.
- Viene usato per passare al modo PRO quando è impostato il modo RUN.
- I modi RUN e PRO vengono selezionati alternativamente ogni volta che premete il tasto [BASIC] .

[SHIFT] [P↔NP]

- Viene usato per impostare il modo stampa e non stampa quando è collegata una stampante facoltativa al **computer**.

[SHIFT] [DRG]
- Viene usato per dare l'esponente.

[hyp]
- Viene usato per indicare un modo angolare.

[SHIFT] [arc hyp]
- Viene usato per scrivere una funzione iperbolica.

[sin] ~ [x^2]
[SHIFT] [$\sin^{-1}$]
[SHIFT] [$n!$]
- Viene usato per scrivere le funzioni iperboliche inverse.

[ENTER]
- Immette una riga di programma nel computer.
- Viene usato per scrivere i programmi.
- Richiede il calcolo manuale o l'esecuzione diretta delle istruzioni di comando da parte del computer.
- Dà un'istruzione di riavviamento dopo aver immesso i dati richiesti da un'istruzione INPUT o dopo aver eseguito un'istruzione PRINT.

Per i tasti impiegati per le operazioni su matrici, consultate pagina 45.

APPENDIEC E
Funzioni dei tasti nel BASIC

I tasti [↑] e [↓] hanno le funzioni seguenti a seconda dei modi scelti, nonché dalla condizione del computer.

Modo	Stato	[↓]	[↑]
RUN	Programma in esecuzione		
	Programma temporaneamente interrotto	Per eseguire la riga successiva	Tenete premuto questo tasto per visualizzare la riga di programma in corso di esecuzione o quella già eseguita.
	Istruzione INPUT in esecuzione		
	Istruzione PRINT appena eseguita		
	Interruzione		
	Condizione di errore durante l'esecuzione di un programma		Tenete premuto questo tasto per visualizzare la riga causa dell'errore.
	Condizione TRON	Per eseguire il debugging (controllo)	Tenete premuto questo tasto per visualizzare la riga di programma in corso di esecuzione o quella già eseguita.
	Altra condizione	Per visualizzare un risultato appena calcolato (risultato dell'ultima funzione)	Come a sinistra
PRO	(Quando il modo viene cambiato da RUN a PRO e la riga di programma non è visualizzata)		
	Programma temporaneamente interrotto	Per visualizzare la riga interrotta	Come a sinistra
	Condizione di errore	Per visualizzare la riga con l'errore	Come a sinistra
	Altra condizione	Per visualizzare la prima riga	Per visualizzare l'ultima riga

Modo	Stato	[↓]	[↑]
PRO	(Quando è visualizzata la riga di programma)	Per visualizzare la riga di programma successiva	Per visualizzare la riga di programma precedente.

Attenzione: I seguenti tasti non possono essere usati in BASIC (modo RUN o PRO).

[FSE], [SHIFT] [TAB], [↕], [SHIFT] [STAT], [SHIFT] [Δ%], [Σ→M], [RM] [M+], [+/−], e i tasti per i calcoli statistici (n, $\bar{x}$, ecc.)

APPENDICE F
CARATTERISTICHE

Modello:	PC-1430 computer tascabile
Processore:	8 bit CMOS CPU
Linguaggio di programmazione:	BASIC
ROM del sistema:	72 K Byte
Capacità di memoria:	RAM:
	Sistema: Circa 1, 1 K Byte
	Utente:
	Area fissa di memoria (A — Z, A$ — Z$) 208 Byte
	Area di programma/dati 6878 Byte
Livelli:	Subroutine: 10 livelli Funzioni: 16 livelli
	FOR–NEXT: 5 livelli Dati: 8 livelli
Operatori:	Addizione, sottrazione, moltiplicazione, divisione, elevamento a potenza, funzioni trigonometriche e trigonometriche inverse, funzioni logaritmiche ed esponenziali, conversione degli angoli, radice quadrata, segno, assoluto, intero, operatori di confronto, operatori logici, operazioni su matrici
Precisione numerica:	10 cifre in mantissa più 2 cifre in esponente.
Funzioni di correzione e controllo:	Cursore destra-sinistra, riga in alto ed in basso, inserimento e cancellazione di caratteri.
Protezione della memoria:	Per mezzo di alimentazione a pile del CMOS anche a macchina spenta.
Visore:	Display a cristalli liquidi, a 24 caratteri, a matrice di 5 × 7 punti
Tasti:	77 tasti: alfabetici, numerici, simboli speciali e funzioni. Tasti definiti dall'utente.
Alimentazione:	6,0 V in C.C., pile al litio, tipo CR-2032 x 2
Consumo:	0,03 W su 6 V C.C.
	Circa 120 ore di funzionamento continuo in condizioni normali (stima basata sull'assunzione di 10 minuti di calcoli o esecuzione di programmi e 50 minuti di esibizione sul visore ogni ora, ad una temperatura di 20° centigradi). La durata delle batterie può tuttavia variare a seconda del particolare uso che viene fatto del computer e del tipo di batterie impiegato.
Temperatura di esercizio:	0°C – 40°C

Dimensioni:	170 (larghezza) x 72 (profondità) x 9,5 (altezza) mm.
Peso:	Circa 150 g (con due batterie ed una scheda RAM)
Accessori di serie:	Astuccio, due pile al litio già inserite, mascherina per la tastiera, manuale di istruzioni. Registratore a cassette (CE-152) Interfaccia stampante/registratore (CE-126P)

APPENDICE G
USO DI PROGRAMMI SCRITTI CON ALTRI MODELLI PC

I programmi scritti sui seguenti modelli di computer PC possono essere usati sul PC-1403, con alcune leggere modifiche.

Serie PC-1210: PC-1210,PC-1211
Serie PC-1245: PC-1245,PC-1246,PC-1247
Serie PC-1250: PC-1250,PC-1251
Serie PC-1260: PC-1260,PC-1261
Serie PC-1350: PC-1350
Serie PC-1401: PC-1401,PC-1402
Serie PC-1450: PC-1450
Serie PC-1460: PC-1460
Serie PC-2500: PC-2500

Benché le funzioni di questi modelli siano tutte diverse, in un modo o nell'altro, i programmi scritti con uno qualsiasi di essi possono essere usati con il PC-1403, eseguendo le modifiche descritte di seguito.

Nota 1 Il PC-1403 può leggere dei programmi da nastri, registrati con i programmi scritti con i computer della serie PC-1210, PC-1245 e PC-1250; tuttavia, i programmi scritti e registrati con il PC-1403 non possono essere letti o usati con i computer di queste tre serie.

Nota 2 I nastri con programmi per i computer della serie PC-1245 e PC-1210 registrati con un numero di programmi caricati con il comado MERGE non possono essere usati sul PC-1403. È necessario eseguire il MERGE dei singoli programmi sul PC-1403.

Nota 3 Programmi contenenti i comandi POKE o CALL, scritti con un PC-1250 non possono essere eseguiti dal PC-1403. Il tentativo di usare questi programmi, potrebbe rendere tutti i tasti del computer inoperativi.

Modificazioni necessarie ai programmi della serie PC-1245 (inclusa la serie PC-1250)

Quando usate i programmi sviluppati per la serie PC-1245 sul PC-1403, è necesario effettuare le seguenti modifiche:

1. **Moltiplicazione senza l'uso dell'operatore " * ":**
   Sulla serie PC-1245, l'operatore (*) per la moltiplicazione può essere omesso, per esempio AB per A * B o CD per C * D. Sul PC-1403, l'operatore di moltiplicazione (*) non può essere omesso poiché il computer tratta due caratteri consecutivi, quali AB o CD, come variabili semplici.

   Esempio A = SIN BC → A = SIN (B * C)

2. **Definizione delle variabili con suffisso (quali A( )) mediante l'uso dell'istruzione DIM:**
   Sulla serie PC-1245, se viene eseguita per esempio l'istruzione DIM A(3Ø), vengono riservate le posizioni di memoria per A(27)–A(3Ø) come estensione di un'area di definizione delle variabili fisse. Viceversa sul PC-1403, l'esecuzione dell'istruzione DIM A(3Ø) riserva un'area di memoria separata per le matrici di variabili A(Ø)–A(3Ø) per la matrice chiamata A.

   Quando definite le variabili con suffisso (quali A( )) come estensione delle variabili fisse, usate la specifica sulla destra dell'esempio seguente:

   DIM A(3Ø) → A(3Ø)=Ø

3. **Istruzione di I/O per i file su nastro:**
   Sulla serie PC-1245, per esempio l'esecuzione dell'istruzione PRINT#C trasferisce il contenuto della variabile C e di tutte le variabili successive in un file su nastro. Viceversa sul PC-1403, l'esecuzione della stessa istruzione trasferisce soltanto il contenuto della variabile C.

   Per trasferire il contenuto di una variabile specifica e di tutte le variabili successive, usate l'istruzione riportata alla destra degli esempi seguenti:

   Esempi PRINT#A → PRINT#A*
   INPUT#C → INPUT#C*

4. **Valore di una variabile di ciclo dopo il completamento di un ciclo FOR-NEXT:**
   Il valore di una variabile di ciclo ottenuta dopo l'esecuzione di un ciclo FOR-NEXT eseguito sul PC-1403 è diverso da quello ottenuto sul PC-1245. Se usate il valore di una variabile di ciclo in un'espressione condizionale di un programma PC-1245, incrementatelo di uno quando lo usate sul PC-1403.

   Esempio:
   ```
   10 FOR I=0 TO 10
   50 NEXT I
   60 IF I=10 THEN 100
   ```
   Modificate il valore di I nella riga 60 come segue:
   ```
   60 IF I=11 THEN 100
   ```
   (Sul PC-1403, il valore di una variabile di ciclo deve essere incrementato del valore di un passo. Il numero dei cicli di esecuzione del ciclo rimane tuttavia uguale).

5. **Ridefinizione [=]**
   Il tasto [=] non funziona come tasto ridefinibile, nel PC-1403. Di conseguenza, nei programmi che richiedono la ridefinizione del tasto dell'uguale, è necessario usare un altro tasto.

   Es.:
   ```
   100 "=" : ······ →  100 "N" : ·····
   ```

6. **Simbolo di esponente "E":**
   Il PC-1403 usa la lettera maiuscola "E" come simbolo di esponente. E' necessario il cambiamento seguente:

   ```
   A=1.234 E 5 → A=1.234E5
   B=E6        → B=1E6
   ```

   Se un programma del PC-1245 viene letto da un file su nastro nel PC-1403, il cambiamento del simbolo di esponente appena descritto, verrà effettuato automaticamente dal PC-1403.

7. Il codice dei caratteri del PC-1245 e parzialmente diverso da quello del PC-1403
Designando i codici seguenti per mezzo della funzione CHR$, cambiare i codici.

Codice dei caratteri	PC-1245	PC-1403
39 (&27)	‾	'
91 (&5B)	√	[
92 (&5C)	¥	\
93 (&5D)	π	]
96 (&6Ø)	IE	‘
250 (&EA)	—(errore)	‾
251 (&FB)	—(errore)	π
252 (&FC)	—(errore)	√

Nota: Come mostrato sopra, il PC-1403 non ha i caratteri IE.

Ulteriori modifiche

1. La serie PC-1245 usa dei numeri di riga che vanno da 1 a 999, mentre il PC-1403 consente di usare dei numeri di riga che vanno da 1 a 65279. Pertanto, il numero di riga usa 3 Byte nella RAM (la serie PC-1245 usa 2 Byte). La modifica viene effettuata automaticamente quando il programma viene caricato da nastro. Esiste la possibilità di un superamento di memoria (ERROR 6) quando si carica o si esegue un programma lungo. Inoltre, quando una singola riga si avvicina agli 8Ø byte, questa modifica può provocare la cancellazione della fine della riga.

2. Se il nastro si ferma o l'allarme interviene nel corso della lettura di un programma dal nastro, con un programma di un computer della serie PC-1245, il computer rimane occupato per uno o due secondi e nel display appaiono due asterischi.
Questo succede perché il computer deve modificare i numeri di riga, come descritto al punto (1) sopra.

Modifiche da apportare ai programmi del PC-1210

Per utilizzare programmi della serie PC-1210 sul PC-1403, questi devono essere modificati analogamente a quanto avviene per i programmi della serie PC-1245 (con eccezione dei punti 2 e 7). Sono inoltre necessarie le seguenti modifiche.

APPENDICE G
Uso di programmi scritti con altri modelli PC

(1) Istruzione IF:

Se, per esempio, si incontra

```
50 IF A>L PRINT "A" (visualizza "A" se A>L)
```

in un programma scritto per la serie PC-1210, questo viene interpretato come:

```
50 IF A> LPRINT "A" (stampa "A" se A>)
```

e dà un errore.

L'errore si verifica perché un comando che non esiste nella serie PC-1210 è previsto invece nel PC-1403.

Per risolvere questo problema inserite un comando THEN nella istruzione IF nel modo seguente.

```
50 IF A>L THEN PRINT "A"
```

(2) Formato specificato nel comando USING

La funzione del comando USING nel PC-1403 e nel PC-1210 differisce nel modo seguente:

Esempio

```
10 A = -123.456
20 PAUSE USING "####.##"; A
30 PAUSE A, USING "####"; A
```

L'esecuzione di questo programma visualizza quanto segue.

* PC-1210/1211		-123.45
	-123	-123
* PC-1403		-123.45
	-123.45	-123

Per l'esecuzione della riga 30 nella serie PC-1210, la visualizzazione sul lato sinistro segue il formato di quanto visualizzato sul lato destro. Nel PC-1403, la visualizzazione segue il formato specificato precedentemente. Questo vale non soltanto per il comando PAUSE, ma anche per i comandi PRINT e LPRINT.

(3) Omissione di ")"

Nella serie PC-1210, è possibile omettere il simbolo ")" che viene immediatamente prima di un tasto [ENTER] o di un: (due punti). Non può essere invece omesso nel PC-1403.

Accertatevi pertanto di aggiungere il simbolo ")" al programma nel caso fosse stato omesso.

(4) Comando PRINT

Il PC-1403 ha un comando PRINT per la visualizzazione e un comando LPRINT per la stampa, ma tutti i comandi PRINT possono essere utilizzati per stampare se è stato specificato PRINT = LPRINT. La serie PC-1210 non ha il comando LPRINT. Per stampare utilizzando i programmi della serie PC-1210, aggiungete PRINT = LPRINT al programma o eseguite manualmente la stampa.

(5) Variabili

Tutte le variabili vengono conservate nella serie PC-1210 quando viene eseguito un comando RUN.

Nel PC-1403, invece, tutte le variabili da A (27) in poi vengono azzerate. Perciò se occorre la necessità di conservare i valori delle variabili anche alla ripartenza del programma, fate partire l'esecuzione del programma con il comando GOTO o con tasti predefiniti.

Modifiche da apportare ai programmi della serie PC-1260

Modifica del codice dei caratteri

Nella serie PC-1260, il carattere corrispondente al codice di carattere 96 (&6Ø) è uno spazio, mentre nel PC-1403 corrisponde a un simbolo di ' (apice singolo). Perciò, se nel comando CHR$ è stato specificato uno spazio utilizzando il codice 96 sostituitelo col codice a 32 ($2Ø).

Comandi CLS e CURSOR

Il PC-1403 non prevede i comandi CLS e CURSOR. Se un programma dovesse contenerli, eliminarli.

Modifiche necessarie alle serie PC-1401, PC-1450 e PC-1460

I programmi scritti per i computer delle serie PC-1401, PC-1450 e PC-1460 possono essere usati senza modifiche sul PC-1403.

Modifiche per i computer della serie PC-1350 e PC-2500

Il PC-1403 non prevede i comandi elencati sotto. I programmi che dovessero contenerli devono essere modificati, per poter essere usati sul PC-1403.

CLS, CURSOR, MEM$, GCURSOR, GPRINT, LINE, POINT, PRESET, PSET, (TEST)

ESEMPI DI PROGRAMMAZIONE

Avete letto le descrizioni delle varie funzioni dell'unità nei capitoli precedenti ed avete pertanto acquisito una certa familiarità con una numero di comandi di programmazione. Tuttavia, per poter effettivamente imparare a sviluppare programmi di applicazione in BASIC, è assolutamente neceaario che sviluppiate ed eseguiate programmi vostri oltre a quelli di questo manuale.

Potete migliorare le vostre doti di pilota guidando o migliorare la vostra tecnica al tennis facendo esercizio, ed allo stesso modo potete raggiungere la piena capacità di programmazione del computer solo provando ad eseguire quanti programmi possibile, a prescindere dalla vostra abilità nell'utilizzare ciascuno di essi.

E inoltre importante che facciate costante riferimento ai programmi sviluppati da altri. In questo capitolo verranno introdotti alcuni esempi di programmazione che impiegano vari comandi BASIC che potranno farvi da guida.

Per rendere più chiare le spiegazioni, negli esempi che il capitolo contiene sono state adottate le seguenti convenzioni:

① LISTE DI PROGRAMMA
Tutte le liste di programma contenute negli esempi di programmazione sono fornite in stampa dalla CE-126P nelle loro dimensioni reali.

② CAPACITÀ DI PROGRAMMA
Al termine di ciascuna lista di programma, le dimensioni del programma stesso sono indicate in byte.

③ STAMPA
Se un programma richiede la stampa, il programma eseguito viene stampato dalla CE-126P in dimensioni reali.

④ CONTENUTO DI MEMORIA
Nella tabella contenente i dati memorizzati di ciascun esempio di programmazione, le variabili con un uso predeterminato sono indicate con tale uso specifico, mentre quelle che non possiedono un uso predeterminato (ad es. quelle memorizzate nell'area di lavoro per ritenere risultati intermedi di un calcolo ecc.) sono indicate con il simbolo "√".

La SHARP CORPORATION e/o le sue sussidiarie non accettano responsabilità nè obblighi per una qualsiasi perdita finanziaria derivata dall'uso di uno degli esempi di programma riportati in questo manuale. Nel servirvi di questi programmi, non dimenticate che possono non essere completamente adatti al vostro caso o non essere precisi quanto occorre. Controllate perciò con attenzione i dati degli esempi di programma di cui volete fare uso e controllate che siano effettivamente quanto cercate. In caso negativo, modificateli prima di eseguirli in modo che rispondano ai vostri fini.

SOMMARIO

Titolo del programma Pagina

- Conversione tra coordinate ortogonali polari 223
- Calcolo dell'area del triangolo 228
- Cerchio osculatore di due cerchi 235
- Gioco dei numeri 239

TITOLO PROGRAMMA: Conversione tra coordinate ortogonali e polari

Questo programma è molto utile per trattare le conversioni tra coordinate ortogonali e polari in tre dimensioni. Dopo aver impostato i dati, si ottengono i risultati sulla base dell'angolo così definito.

■ PROCEDURA OPERATIVA

1. [DEF] [A] (Conversione di coordinate polari in coordinate ortogonali)
   Impostando le coordinate x, y e z, i valori r, θ e φ vengono visualizzati nel display e il programma è terminato.

2. [DEF] [B] (Conversione di coordinate ortogonali in coordinate polari)
   Impostando le coordinate r, θ e φ, nel display vengono visualizzati i valori x, y e z e il programma è terminato.

Nota – Se l'angolo viene definito in DEG, si usano i gradi sia per l'input sia per l'output; se è invece definito in RAD, si usano le radianti.

■ CONTENUTO

1. Conversione di coordinate ortogonali in coordinate polari con $r = y = 0$, $r = 0$ e θ = indefinito

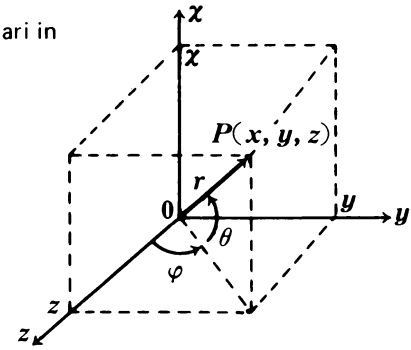
   $r = \sqrt{x^2 + y^2 + z^2}$

   $\theta = \text{sen}^{-1}(z/r)$

   Se $x > 0$, $\varphi = \tan^{-1}(y/x)$
   Se $x = 0$, e $y \geqq 0$, $\varphi = 90°$
   e $y < 0$, $\varphi = 90°$
   Se $x < 0$, e $y \geqq 0$, $\varphi = \tan^{-1}(y/x) + 180°$
   e $y < 0$, $\varphi = \tan^{-1}(y/x) - 180°$

2. Conversione di coordinate polari in coordinate ortogonali

   $x = r \cos\theta \cdot \cos\varphi$
   $y = r \cos\theta \cdot \text{sen}\,\varphi$
   $z = r\,\text{sen}\,\theta$



■ ESEMPIO

1. Conversione di coordinate ortogonali in coordinate polari

$x = -1$
$y = 2$
$z = -3$

Angolo definito in DEG

2. Conversione di coordinate polari in coordinate ortogonali

$r = 3.741657387$
$\theta = -53.30077479$
$\varphi = 116.5650512$

Angolo definito in DEG

■ SEQUENZA OPERATIVA

< Coordinate ortogonali → Coordinate polari >

1. [DEF] [A]

x =_

2. −1 [ENTER]

y =_

3. 2 [ENTER]

z =_

4. −3 [ENTER]

r

5. [ENTER]

3.741657387

6. [ENTER]

THETA

7. [ENTER]

−53.30077479

8. [ENTER]

PHI

9. [ENTER]

116.5650512

10. [ENTER]

〉

< Coordinate polari → Coordinate ortogonali >

1. [DEF] [B]

r =_

2. 3.741657387 [ENTER]

THETA=_

3. −53.30077479 [ENTER]

PHI=_

4. 116.5650512 [ENTER]

x

5. [ENTER]

−1.000000001

6. [ENTER]

y

7. [ENTER]

2.

8. [ENTER]

z

9. [ENTER]

−3.

10. [ENTER]

〉

■ LISTATO

```
 10:"A":INPUT "x=";X
 20:INPUT "y=";Y
 30:INPUT "z=";Z
 40:R=SQR (X*X+Y*Y+Z*Z):
    IF R=0 PAUSE "r=0 Un
    defined":END
 50:C=ASN (Z/R):IF X>0
    LET F=ATN (Y/X):GOTO
    80
 60:A=(Y=0)+SGN Y:IF X=0
    LET F=A*ACS 0:GOTO 8
    0
 70:F=ATN (Y/X)+A*ACS -1
 80:WAIT :PRINT "r":
    PRINT R:PRINT "THETA
    ":PRINT C:PRINT "PHI
    ":PRINT F:END
 90:"B":INPUT "r=";R
100:INPUT "THETA=";C
110:INPUT "PHI=";F
120:Y=R*COS C:X=Y*COS F:
    Y=Y*SIN F:Z=R*SIN C
130:WAIT :PRINT "x":
    PRINT X:PRINT "y":
    PRINT Y:PRINT "z":
    PRINT Z:END
```

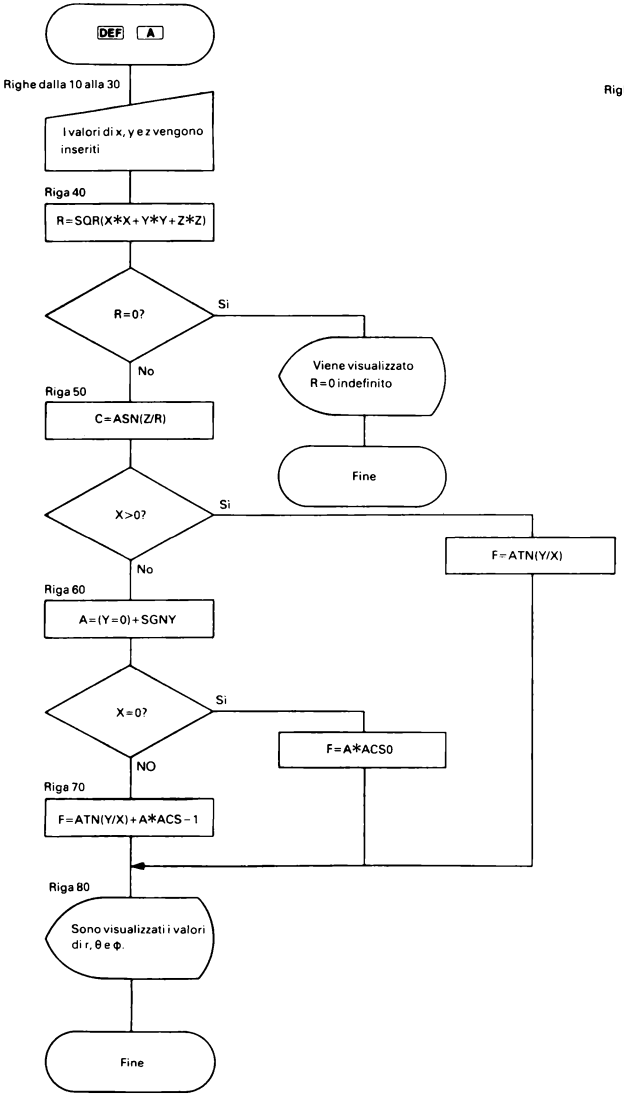
300 Bytes

■ CONTENUTO MEMORIA

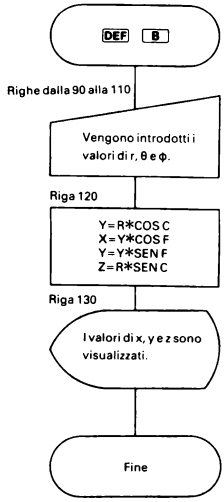
A	$\sqrt{}$
C	θ
F	φ
R	r
X	coordinata x
Y	coordinata y
Z	coordinata z

■ DIAGRAMMA DI FLUSSO

(Coordinate ortogonali → coordinate polari)



(Coordinate polari → coordinate ortogonali)



TITOLO PROGRAMMA: Calcolo dell'area del triangolo Periferiche richieste: CE-126P

Qualsiasi poligono consta di triangoli. Partendo da questo principio, possiamo calcolare l'area di un poligono. Per mezzo di questo programma si può determinare l'area di un poligono come somma delle aree di ogni triangolo di cui il poligono è composto.

■ PROCEDURA OPERATIVA

1. [DEF] [A] (inizio programma)
   Impostate il numero di vertici e le coordinate ad essi corrispondenti.
2. Premete [DEF] [B]
   Impostando i dati relativi a tre vertici, l'area del triangolo, visualizzata nel display, viene stampata. Se, invece di impostare i dati relativi ad un nuovo vertice, si preme il tasto [ENTER] , la stampante stampa l'area totale.
3. Premendo i tasti [DEF] [C] l'area totale viene cancellata e il programma è terminato.

Nota— Il numero massimo di vertici usabili è: 255 vertici

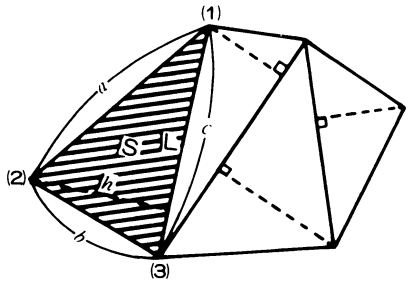
■ CONTENUTO

$$S = \sqrt{s(s-a)(s-b)(s-c)}$$

$$s = \frac{a+b+c}{2}$$

Area del triangolo

$$S = \frac{1}{2} h L$$

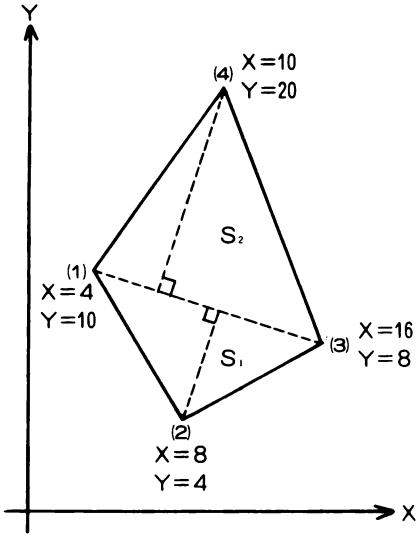


L: Il più lungo dei lati a, b e c.

h: Tralasciate le cifre dopo la quarta dopo la virgola.

■ ESEMPIO

Area di un poligono



■ STAMPA

```
Point 1x= 4
Point 1y= 10
Point 2x= 8
Point 2y= 4
Point 3x= 16
Point 3y= 8
Point 4x= 10
Point 4y= 20
  1-2-3
A=            7.211
B=            8.944
C=           12.166
H=            5.260
S=           31.996580
  1-3-4
A=           12.166
B=           13.416
C=           11.662
H=            9.838
S=           65.993304

*TS*=        97.989884
```

■ SEQUENZA OPERATIVA DEI TASTI

< Impostazione delle coordinate >

1. [DEF] [A]

Numbers=_

2. 4 [ENTER]

Point 1x=
?

3. 4 [ENTER]

Point 1y=
?

Impostazione come sopra

4. 20 [ENTER]

〉

< Impostazione dei numeri delle coordinate >

1. [DEF] [B]

Point=_

2. 1 [ENTER]

Point=_

3. 2 [ENTER]

Point=_

4. 3 [ENTER]

Point=_

5. 1 [ENTER]

Point=_

6. 3 [ENTER]

Point=_

7. 4 [ENTER]

Point=_

8. [ENTER]

〉

< Cancellazione >

9. [DEF] [C]

**** TS CLEAR ****
〉

■ LISTATO DEL PROGRAMMA

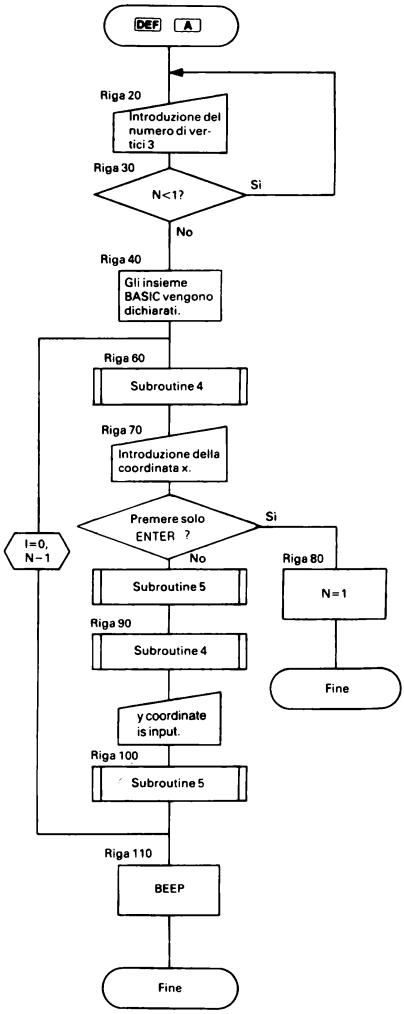
```
 10:"A":USING :CLEAR :
    WAIT 0
 20:INPUT "Numbers=";N
 30:IF N<1 BEEP 2:GOTO 2
    0
 40:DIM X(N-1),Y(N-1),B$
    (0)
 50:FOR I=0 TO N-1
 60:B$(0)="x=":GOSUB 360
 70:INPUT X(I):B$(0)="x=
     "+STR$ X(I):GOSUB 3
    70:GOTO 90
 80:N=I:END
 90:B$(0)="y=":GOSUB 360
    :INPUT Y(I)
100:B$(0)="y= "+STR$ Y(I
    ):GOSUB 370:NEXT I
110:BEEP 1:END
120:"B":USING :INPUT "Po
    int=";O,"Point=";P,"
    Point=";Q:GOTO 140
130:GOTO 310
140:IF (O<1)+(O>N)+(P<1)
    +(P>N)+(Q<1)+(Q>N)<>
    0 GOTO 120
150:C=X(O-1):D=Y(O-1):E=
    X(P-1):F=Y(P-1):G=X(
    Q-1):H=Y(Q-1)
160:X=E-C:Y=F-D:GOSUB 33
    0
170:A=X:X=G-E:Y=H-F:
    GOSUB 330
180:B=X:X=C-G:Y=D-H:
    GOSUB 330
190:C=X:IF A>X LET X=A
200:IF B>X LET X=B
210:I=(A+B+C)/2:S=SQR (I
    *(I-A)*(I-B)*(I-C))
220:J= INT ((2*S/X)*10^3
    )/10^3:L=X:GOSUB 340
230:X=L:S=X*J/2:K=K+S:L=
    A:GOSUB 340
240:A=L:L=B:GOSUB 340
250:B=L:L=C:GOSUB 340
260:C=L:L=S:GOSUB 350
270:S=L:L=K:GOSUB 350
280:K=L:LPRINT "  ";STR$
    O+"-"+STR$ P+"-"+
    STR$ Q
290:LPRINT "A= "; USING
    "##########.###";A
300:LPRINT "B= ";B:
    LPRINT "C= ";C:
    LPRINT "H= ";J:
    LPRINT USING "######
    ######.######";"S= "
    ;S:GOTO 120
310:LPRINT "":LPRINT "*T
    S*="; USING "#######
    ##.######";K:END
320:"C":K=0:USING :PAUSE
    " ** TS CLEAR **":
    END
330:X=SQR (X*X+Y*Y):
    RETURN
340:L= INT (L*1000+.5)/1
    000:RETURN
350:L= INT (L*1000000)/1
    000000:RETURN
360:PAUSE "Point ";STR$
    (I+1);B$(0):RETURN
370:LPRINT "Point ";STR$
    (I+1);B$(0):RETURN
```

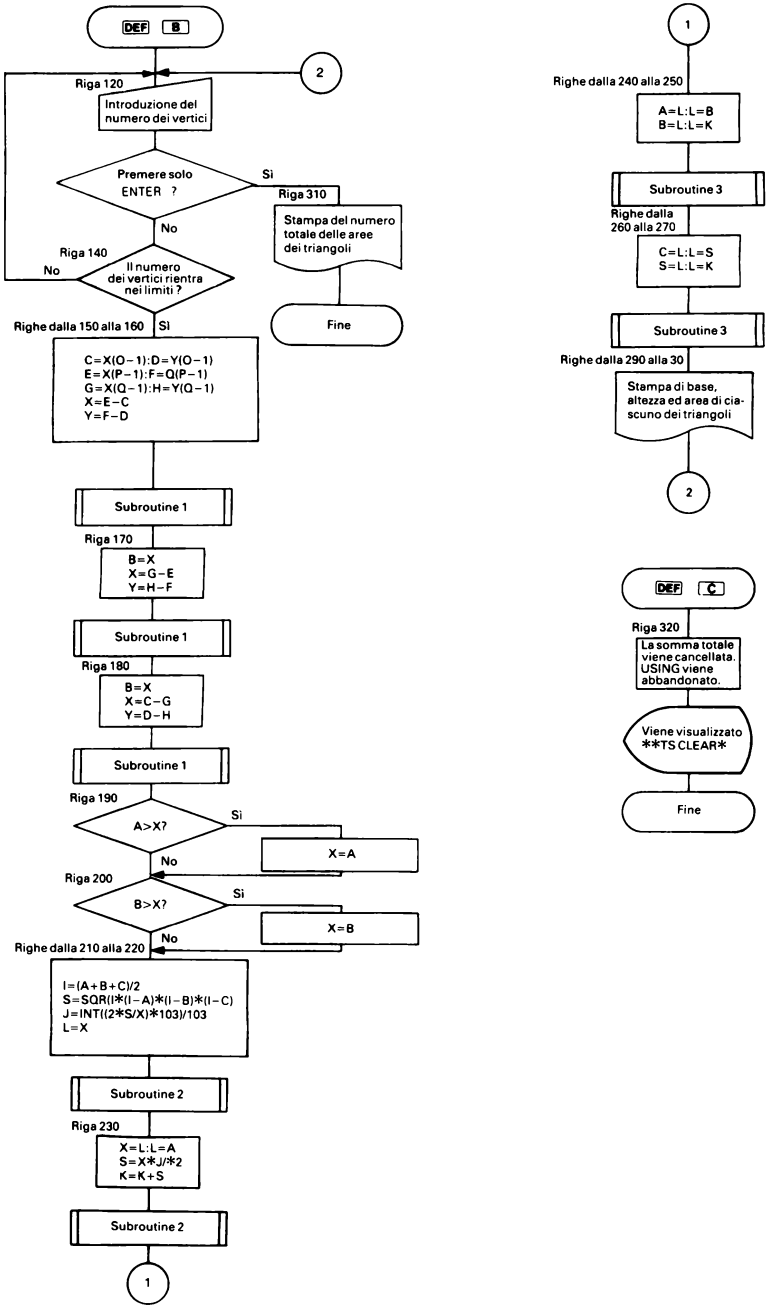
962 Bytes

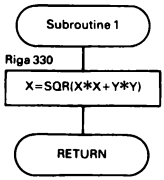
■ CONTENUTO DELLA MEMORIA

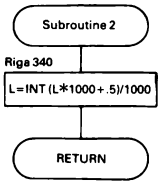
A	a
B	b
C	x_1, √
D	y_1
E	x_2
F	y_2
G	x_3
H	y_3
I	S
J	h
K	Σs
L	√
N	numero di vertici
O	√
P	√
Q	√
S	S
X	√
Y	√
X(N-1)	coordinata x
Y(N-1)	coordinata y
B$($\varphi$)	√

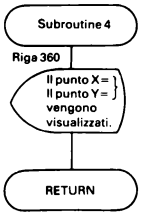
■ DIAGRAMMA DI FLUSSO

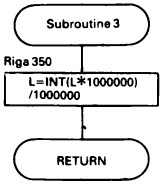


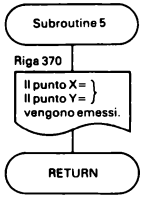

DEF B
2
Riga 120
Introduzione del numero dei vertici
Premere solo ENTER ?
Si
Riga 310
Stampa del numero totale delle aree dei triangoli
Fine
No
Riga 140
No
Il numero dei vertici rientra nei limiti ?
Righe dalla 150 alla 160
Si
C=X(O-1):D=Y(O-1)
E=X(P-1):F=Q(P-1)
G=X(Q-1):H=Y(Q-1)
X=E-C
Y=F-D
Subroutine 1
Riga 170
B=X
X=G-E
Y=H-F
Subroutine 1
Riga 180
B=X
X=C-G
Y=D-H
Subroutine 1
Riga 190
A>X?
Si
X=A
No
Riga 200
B>X?
Si
X=B
No
Righe dalla 210 alla 220
I=(A+B+C)/2
S=SQR(I*(I-A)*(I-B)*(I-C)
J=INT((2*S/X)*103)/103
L=X
Subroutine 2
Riga 230
X=L:L=A
S=X*J/*2
K=K+S
Subroutine 2
1
1
Righe dalla 240 alla 250
A=L:L=B
B=L:L=K
Subroutine 3
Righe dalla 260 alla 270
C=L:L=S
S=L:L=K
Subroutine 3
Righe dalla 290 alla 30
Stampa di base, altezza ed area di ciascuno dei triangoli
2
DEF C
Riga 320
La somma totale viene cancellata. USING viene abbandonato.
Viene visualizzato **TS CLEAR*
Fine


Subroutine 1
Riga 330
X=SQR(X*X+Y*Y)
RETURN


Subroutine 2
Riga 340
L=INT (L*1000+.5)/1000
RETURN


Subroutine 4
Riga 360
Il punto X=
Il punto Y=
vengono
visualizzati.
RETURN


Subroutine 3
Riga 350
L=INT(L*1000000)
/1000000
RETURN

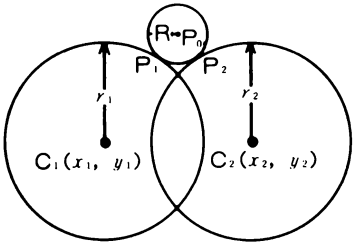

Subroutine 5
Riga 370
Il punto X=
Il punto Y=
vengono emessi.
RETURN

TITOLO PROGRAMMA: Cerchio osculatore di due cerchi

Vicino a due cerchi, se ne trova un terzo. Questo programma è stato concepito per determinare le coordinate al centro del cerchio che oscula gli altri due cerchi, e le coordinate dei punti di tangenza dei due cerchi, impostando le coordinate del centro dei due cerchi ed il raggio, insieme con tre condizioni discriminatorie.

■ PROCEDURA OPERATIVA

1. [DEF] [A] (inizio programma)
2. Impostate le coordinate (x_1, y_1) del centro del cerchio C_1 ed il raggio r_1; le coordinate (x_2, y_2) del centro del cerchio C_2 ed il raggio r_2; e il raggio R del cerchio osculatore dei cerchi C_1 e C_2.



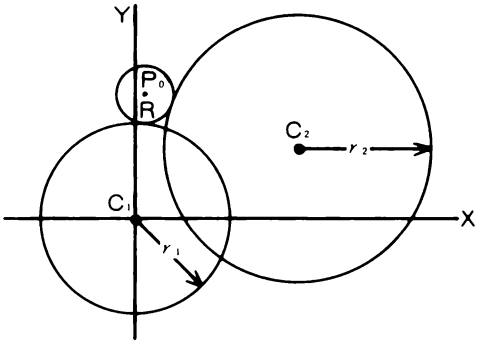
3. Impostate quindi i dati delle condizioni discriminatorie.

Condizione discriminatoria

(1)	Cerchio rispetto al quale il cerchio C_1 è tangente esternamente.	1
	Cerchio rispetto al quale il cerchio C_1 è tangente internamente.	−1
(2)	Cerchio rispetto al quale il cerchio C_2 è tangente esternamente.	1
	Cerchio rispetto al quale il cerchio C_2 è tangente internamente.	−1
(3)	Osservazione del cerchio C_2, dal cerchio C_1, mentre il cerchio si trova a sinistra	1
	Osservazione del cerchio C_2, dal cerchio C_1, mentre il cerchio si trova a destra	−1

Dopo aver impostato i dati di cui sopra, nel display vengono visualizzate le coordinate del centro del centro osculante C_1 e C_2 e le coordinate dei punti di tangenza P_1 e P_2. Fine programma.

■ ESEMPIO



C_1: $x_1 = 0$, $y_1 = 0$, $r_1 = 30$
C_2: $x_2 = 50$, $y_2 = 20$, $r_2 = 40$
R = 10

Condizione discriminatoria
(1) 1 (tangente esternamente)
(2) 1 (tangente esternamente)
(3) 1 (lato sinistro)

■ SEQUENZA OPERATIVA DEI TASTI

1. [DEF] [A]

C1 x=_

2. 0 [ENTER]

C1 y=_

3. 0 [ENTER]

C1 r=_

⋮

Impostazione come sopra

⋮

4. 10 [ENTER]

C1 . OUT:1 IN:−1=_

5. 1 [ENTER]

C2 . OUT : 1 IN :−1=_

6. 1 [ENTER]

LEFT : 1RIGHT : 1−1=_

7. 1 [ENTER]

P0 x= 4.08

8. [ENTER]

P0 y= 39.79

9. [ENTER]

P1 x= 3.06

10. [ENTER]

P1 y= 29.84

11. [ENTER]

P2 x= 13.27

12. [ENTER]

P2 y= 35.83

13. [ENTER]

>

■ LISTATO DEL PROGRAMMA

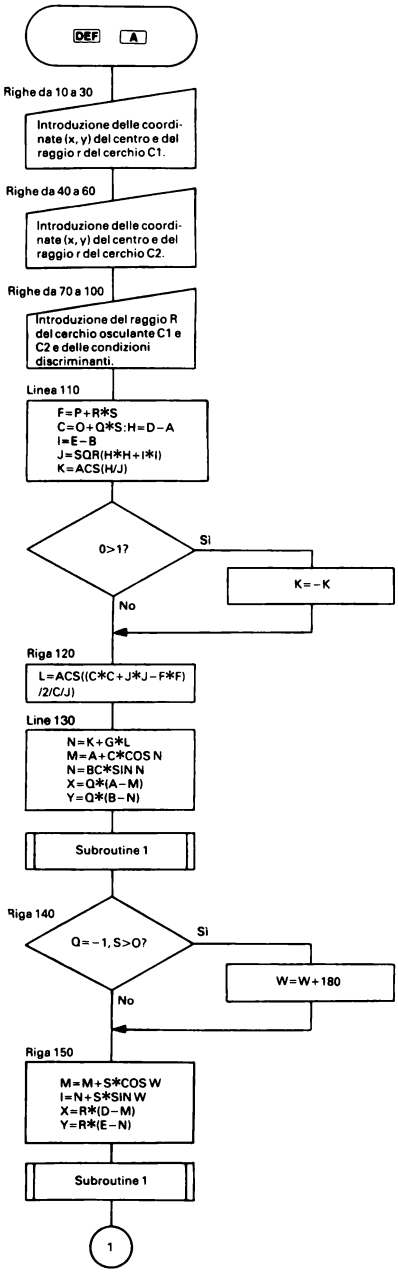
```
 10:"A":DEGREE :INPUT "C
    1 x=";A
 20:INPUT "C1 y=";B
 30:INPUT "C1 r=";O
 40:INPUT "C2 x=";D
 50:INPUT "C2 y=";E
 60:INPUT "C2 r=";P
 70:INPUT "R=";S
 80:INPUT "C1.OUT:1 IN:-
    1=";Q
 90:INPUT "C2.OUT:1 IN:-
    1=";R
100:INPUT "LEFT:1 RIGHT:
    -1=";G
110:F=P+R*S:C=O+Q*S:H=D-
    A:I=E-B:J=SQR (H*H+I
    *I):K=ACS (H/J):IF 0
    >I LET K=-K
120:L=ACS ((C*C+J*J-F*F)
    /2/C/J)
130:N=K+G*L:M=A+C*COS N:
    N=B+C*SIN N:X=Q*(A-M
    ):Y=Q*(B-N):GOSUB 24
    0
140:IF ((Q=-1)*(S>0))=1
    LET W=W+180
150:H=M+S*COS W:I=N+S*
    SIN W:X=R*(D-M):Y=R*
    (E-N):GOSUB 240
160:IF ((R=-1)*(S>P))=1
    LET W=W+180
170:J=M+S*COS W:K=N+S*
    SIN W
180:M=M+SGN M*.005:N=N+
    SGN N*.005
190:H=H+SGN H*.005:I=I+
    SGN I*.005
200:J=J+SGN J*.005:K=K+
    SGN K*.005
210:PRINT "P0 x=";USING
    "########.##";M:
    PRINT "P0 y=";N
220:PRINT "P1 x=";H:
    PRINT "P1 y=";I
230:PRINT "P2 x=";J:
    PRINT "P2 y=";K:END
240:W=ACS (X/SQR (X*X+Y*
    Y)):IF 0>Y LET W=-W
250:RETURN
```

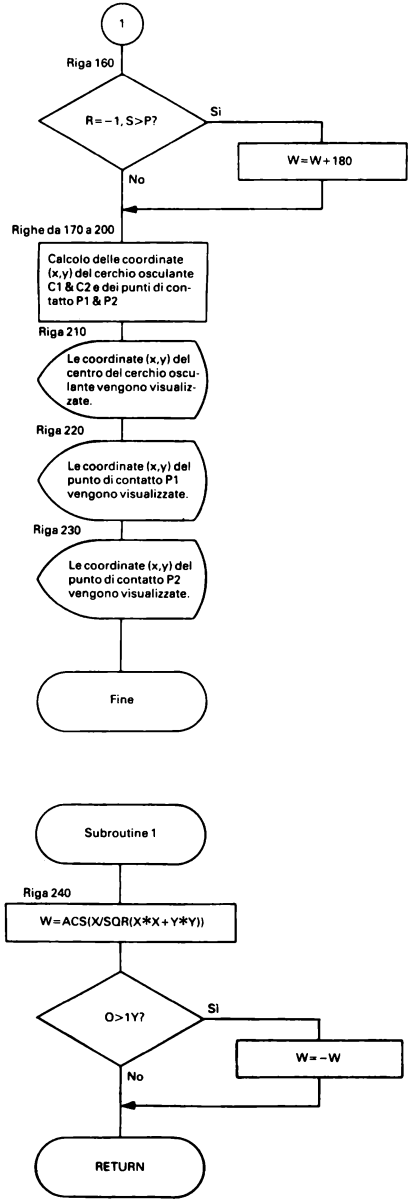
653 Bytes

■ MEMORY CONTENTS

A	x_1
B	y_1
C	√
D	x_2
E	y_2
F	√
G	Discriminating condition (3)
H	P_1x
I	P_1y
J	P_2x
K	P_2y
L	√
M	P_0x
N	P_0y
O	r_1
P	r_2
Q	Discriminating condition (1)
R	Discriminating condition (2)
S	R
W	√
X	√
Y	√

■ DIAGRAMMA DI FLUSSO





TITOLO DEL PROGRAMMA: Gioco dei numeri

Questo programma è un gioco, con il quale si devono indovinare dei numeri di tre cifre generati a caso dal computer. Non studiate troppo! A volte, tentate la fortuna. Indovinate. Provate a vedere quanti numeri sapete indovinare.

■ SVILUPPO DEL GIOCO

1. [DEF] [A] (inizio programma)
2. Nel display viene visualizzato X. Battete il numero che pensate il computer abbia preparato. Al che il computer visualizza il vostro numero, il numero del tentativo che avete eseguito ed un commento. Tempo a disposizione: 1 secondo.

   Esempio:

   • Commento: 1 1

   Nel caso della visualizzazione di cui sopra, il primo numero 1 significa che una delle cifre che avete impostato è corretto e che anche la posizione è corretta. Il secondo 1 indica che una delle cifre che avete impostato è corretta ma no si trova nella posizione giusta.

   • Commento: 3 0

   Nel caso della visualizzazione di cui sopra, tutte le cifre e tutte le posizioni sono corrette. Dopo aver visualizzato il messaggio ed il numero di tentativi che avete fatto, il programma si conclude.

Nota— Il numero generato dal computer è un numero a tre cifre.

■ SEQUENZA OPERATIVA DEI TASTI

1. [DEF] [A]

X=_

2. 123 [ENTER]

1	123
Comment:0	1
X=_	

3. 145 [ENTER]

2	145
Comment:1	0
X=_	

Impostazione come sopra

4. 305 [ENTER]

6	305
Comment:3	0
VERY GOOD !	6
>	

■ LISTATO DEL PROGRAMMA

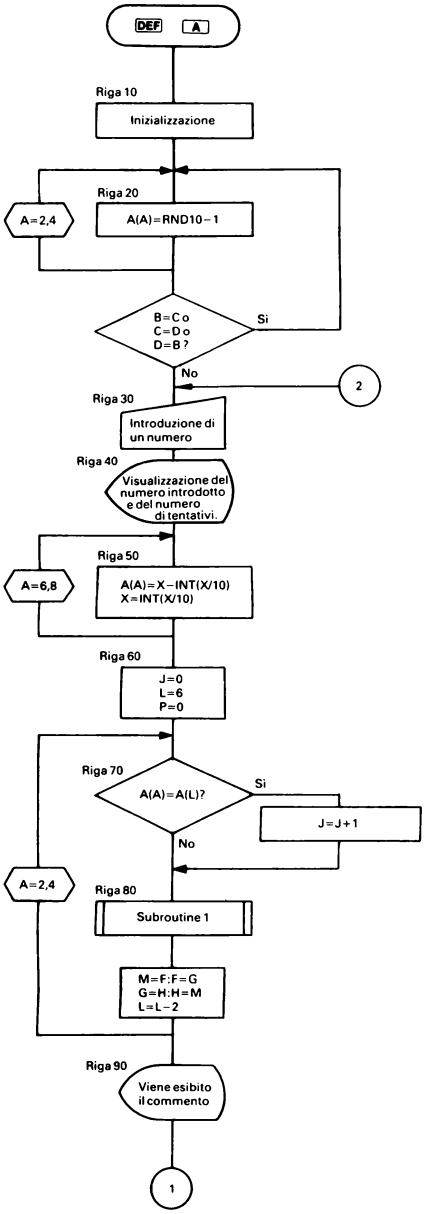
```
 10:"A":CLEAR :RANDOM :Y
    =1
 20:FOR A=2 TO 4:A(A)=
    RND 10-1:NEXT A:IF (
    B=C)+(C=D)+(D=B)<>0
    THEN 20
 30:BEEP 1:INPUT "X=";X
 40:USING :PAUSE Y,X
 50:FOR A=6 TO 8:A(A)=X-
    INT (X/10)*10:X= INT
    (X/10):NEXT A
 60:J=0:L=6:P=0
 70:FOR A=2 TO 4:IF A(A)
    =A(L) LET J=J+1
 80:GOSUB 110:GOSUB 110:
    M=F:F=G:G=H:H=M:L=L-
    2:NEXT A
 90:PAUSE "Comment:";
    USING "####";J;P:IF
    J<>3 LET Y=Y+1:GOTO
    30
100:BEEP 2:PRINT "VERY G
    OOD ! ";Y:END
110:L=L+1:IF A(A)=A(L)
    LET P=P+1
120:RETURN
```

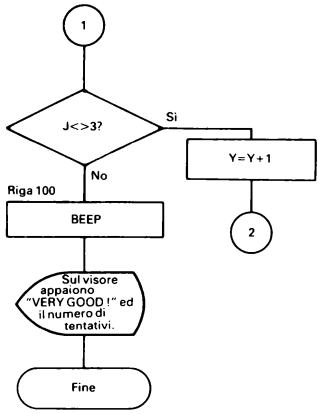
309 Bytes

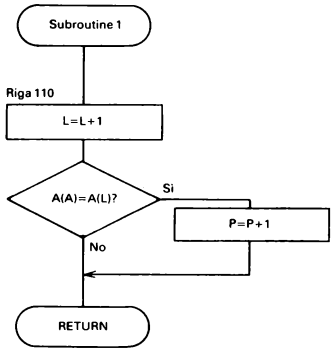
■ CONTENUTO DELLA MEMORIA

A	√
B	√
C	numero di tre cifre
D	√
F	√
G	√
H	√
J	commento
L	√
M	√
P	commento
X	impostazione valore
Y	tentativi eseguiti

■ DIAGRAMMA DI FLUSSO







INDICE

& 78
* 87
+ 87
− 87
/ 87
^ 87
√ 188
< 88
◀ 57
<= 88
<> 88
= 88
> 88
▶ 57
>= 88
π 182
↑ 96
↓ 96
ABS 183
ACS 183
AHC 183
AHS 183
AHT 183
ALL RESET 9
AND 89
AREAD 134
ASC 226
ASN 184
Astuccio rigido 3
ATN 184
BEEP 136
Calcoli di revisione 57
Calcoli statistica a due variabili 37
Calcoli statistici con una variabile 35
Calcoli statistico 34
Calcolo manuale 56
CE-126P 105, 157
CHAIN 137
CHR$ 189
CLEAR 139
CLOAD 117
CLOAD? 119
Comandi 94, 114, 117
CONT 120
Controllo dei programmi 192
COS 184
CSAVE 121
CUR 184
Cursore 8
DATA 140
DEG 184
DEGREE 141
DELETE 122
DIM 142
DMS 185
END 144
Espressioni 87
Espressioni di relazione 88
Espressioni logiche 89
Espressioni numeriche 87
Espressioni stringa 88
EXP 185
FACT 185
FOR...TO...STEP 145
Formati di visualizzazione 200
Funzione calcolo diretto 76
Funzione ultimo risultato 69
Funzioni 92
Funzioni numeriche 183
Funzioni stringa 189
Gamma dei numeri 68
GOSUB 147
GOTO 123, 148
GRAD 149
HCS 185
HSN 185
HTN 185
IF...THEN 150
INKEY$ 181
INPUT 151

INPUT # 153
INT 186
Istruzioni 93
LEFT$ 189
LEN 189
LET 156
Limiti dei numeri 68
LIST 124
LLIST 125
LN 186
LOG 186
LPRINT 157
Manutenzione 194
Maschera 200
Mascherina 104
Matrici di variabili 82
MDF 159
MEM 182
MERGE 126
Messaggi di errore 195
MID$ 190
Modo BASIC 7
Modo CAL 7
Modo PROgram 6
Modo RUN 92
Nastro a cassetta 112
NEW 129
NEXT 160
NOT 89
Notazione scientifica 67
Numero di riga 93
ON (Accensione) 13
ON...GOSUB 161
ON...SOTO 162
OR 89
Operatori 87
Operazioni su matrici MATRIX 44
P NP 106
Parentesi 91
PASS 130
PAUSE 163
PI 182
POL 186
PRINT 165
PRINT # 167
Priorità (modo Cal) 28
Priorità degli operatori (modo BASIC) 204
Programmi 93
Programmi etichettati 103
Protezione della memoria 130
Pseudovariabili 181
RADIAN 169
RANDOM 170
RCP 186
READ 171
REC 186
Registratore a cassette 107
Regressione lineare 37
Revisione del programma 96
REM 172
RENUM 131
RESET 9
RESTORE 173
RETURN 174
Ricerca dei guasti 191
RIGHT$ 190
RND 187
ROT 187
RUN 133
SGN 187
SIN 188
Sostutuzione delle batterie 10
Spegnimento automatico (Auto Off) 13
SQR 188
SQU 188
Stampante 105
STOP 175
STR$ 190
Subroutine 147
TAN 188
Tasto BASIC 6
Tasto C.CE 206
Tasto CA 206
Tasto CAL 6

Tasto DEF 103
Tasto DELete 97
Tasto ENTER 57
Tasto INSert 59
Tasto SHIFT 14
Tasto di cancellazione 15
Tavola dei codici ASCII 198
TEN 188
TROFF 176
TRON 177
USING 178
VAL 190
Variabili 79
Variabili A( ) 85
Variabili fisse 80
Variabili numeriche 80
Variabili semplici 81
Variabili stringa 80
Verbi 94, 115, 134
Visore 18
WAIT 180

SHARP CORPORATION

OSAKA, JAPAN

1986 © SHARP CORPORATION

PRINTED IN JAPAN/IMPRIMÉ AU JAPON
6G2T(TINSI1096ECZZ)